

Projektgruppe OLIVANDA
Abschlussbericht

Studierende des Departments für Informatik,
Carl-von-Ossietzky Universität Oldenburg

Inhaltsverzeichnis

Teil I.

Abschlussbericht

1. Einleitung

OLIVANDA — OLdenburgs Interactive VirtuAl Networked meDiA — ist eine Projektgruppe des Departments für Informatik an der Carl-von-Ossietzky Universität in Oldenburg. Sie ist angesiedelt in der Abteilung Medieninformatik unter der Leitung von Prof. Dr. Susanne Boll und in Kooperation mit dem Oldenburger Forschungs- und Entwicklungsinstitut für Informatik-Werkzeuge und Systeme.

Die Projektgruppe besteht aus einem neunköpfigen Entwicklungsteam, welches von den OFFIS-Mitarbeitern Martin Pielot, Niels Henze und Daniel Wichmann betreut wird. Im Zeitraum vom 1.10.2007 bis zum 30.09.2008 wurde in einem agilen Softwareentwicklungsprozess eine mobile Anwendung zur Verknüpfung realer Objekte mit digitalen Repräsentationen erstellt.

Als Vorgehensmodell wurde Extreme Programming (im Folgenden „XP“ [1] genannt) angewendet. Dieses von Kent Beck entwickelte Verfahren setzt das „Agile Manifest“ um und beruht auf verschiedenen Techniken, wie bspw.

Pair-Programming: Programmieren in Zweierteams an einem Rechner, wodurch ein ständiger Codereview und eine schnelle Fehlerauffindung erreicht wird;

Test-driven development: Zuerst wird ein Test geschrieben, erst danach folgt der Code, der diesen Test erfüllt;

Collective Ownership: Jeder Entwickler ist für den gesamten Programmcode verantwortlich.

Dabei folgt die Softwareentwicklung nicht einem starren Entwurf, sondern lediglich einem Leitszenario. In enger Zusammenarbeit mit dem Auftraggeber werden Anwendungsmöglichkeiten auf sogenannten Storycards („SCs“) ausformuliert, wobei die technische Umsetzung auf überschaubare Teilfunktionen („Tasks“) aufgegliedert wird. Der genaue Verlauf des Projektes mit XP wird im zweiten Teil des Abschlussberichtes vorgestellt.

Vor dem Hintergrund des OFFIS-Projekts „Intermedia“ wurde eine „Mobile-Client to Server“-Anwendung auf J2ME-Basis entwickelt. Ziel der Entwicklung war es, Plakate zu erkennen und dem Nutzer plakatbezogene, personalisierte sowie kontextbezogene Dienste zur Verfügung zu stellen. Als Anwendungsbeispiel sei hier das Fotografieren eines Kinoplakats durch den Benutzer genannt, woraufhin er das Bild an den Dienst verschickt

[1] http://www.bitpipe.com/tlist/Software-Development-Tools.html

und beispielsweise den Trailer gesendet bekommt oder mit einem Kartenbestellservice verbunden wird.

Dieser Bericht fasst die Arbeit der Projektgruppe zusammen und gibt einen Überblick über die im Rahmen der Projektarbeit angefallenen Probleme und ihre Lösungen. Dies ist in Teil 1 des Berichts zu finden. Die ersten beiden Kapitel widmen sich der Einführung in die Thematik, der Struktur der Projektgruppe und der konkreten Zielsetzung. Anschließend wird im dritten Kapitel beschrieben, wie die festgelegten Ziele konzeptuell erreicht werden sollen. Die Kapitel 4 und 5 behandeln die beiden Evaluationen, die ihm Rahmen der Projektgruppe durchgeführt wurden und im sechsten und letzten Kapitel des ersten Teils wird ein Fazit des Projekts gezogen.

Der zweite Teil des Berichts behandelt den Verlauf der Softwareentwicklung im Extreme Programming. Nach einer kurzen Einleitung im Kapitel 7 werden in den vier folgenden Kapiteln die einzelnen Releases des XP dokumentiert. Das Zwölfte Kapitel beinhaltet eine Installationsanleitung für Server- und Clientsoftware.

Aus Gründen der besseren Lesbarkeit verwenden wir innerhalb der gesamten Projektdokumentation das generische Maskulinum.

1.1. Motivation

Das Ziel der Projektgruppe war, eine stärkere Verknüpfung zwischen Objekten aus der realen Welt mit deren Repräsentationen in der virtuellen Welt herzustellen. So kann der Nutzer ein Kinoplakat mit seinem Handy fotografieren und erhält daraufhin zusätzliche Informationen, wie zum Beispiel den Ort des nächstgelegenen Kinos, oder aber Informationen über die Schauspieler und eine Beschreibung des Films. Aus eigener Erfahrung zeigt sich, dass man, während man unterwegs ist, oftmals interessante Plakate wahrnimmt, deren Inhalt man aber kurze Zeit später wieder vergessen hat. Daher möchten wir diesen Nutzern durch den Einsatz kontextueller Bookmarks die Möglichkeit geben, solche Informationen kontextuell auf ihren Handys zu speichern.

So werden durch die Nutzung von Olivanda klassische Werbemedien informativer und interessanter, was auch für Werbeunternehmen eine gute Möglichkeit ist, ihre Produkte in der relevanten Zielgruppe besser zu positionieren. In Kooperation mit der „Ströer Out-of-Home Media“, einem auf Außenwerbung spezialisierten Unternehmen, und dem Lehrstuhl Marketing wurde daher im Sommersemester 2008 eine Lehrveranstaltung angeboten, die die Konzeption einer Plakatkampagne in Verbindung mit den Möglichkeiten von Olivanda verbindet.

Zur besseren Bewertung der Nützlichkeit des Projekts wurde nach Fertigstellung des ersten Prototyps eine Evaluation des Olivanda-Systems durchgeführt. Eine weitere Evaluation folgte kurz darauf im Zusammenhang der Kooperation mit Ströer Media (siehe Abschnitt 1.5). Die Evaluationen werden im Kapitel 4 gesondert beschrieben und es wird auf ihre Ergebnisse eingegangen.

1.2. Related Work

Die Idee, über Bilder eine Verknüpfung zwischen der digitalen und der realen Welt herzustellen haben sich auch andere Systeme zunutze gemacht.

Von der Firma Vodafone wurde auf der CeBIT 2008 die Bildsuchmaschine Otello[2] vorgestellt. Der Nutzer kann ein Bild mit seinem Handy machen und dieses per MMS an eine Service-Nummer von Vodafone schicken um Informationen zum fotografierten Objekt oder Serviceleistungen wie beispielsweise einen Songdownload zu erhalten. Zudem sieht das Programm eine Annotation unbekannter Objekte durch den Benutzer vor. Wenn das fotografierte Bild nicht in der Vodafone-Datenbank gefunden wird, kann der Nutzer es dort einstellen und selbst Informationen dazu verfassen. Ein Nutzer kann sich dort Profile anlegen oder bestimmten Gruppen beitreten ähnlich dem StudiVZ. Vodafone verwendet zur Übertragung der Informationen die MMS-Technologie. Die Bilderkennung wird, wie beim Olivanda-System, durch den SIFT-Algorithmus durchgeführt.

Die Universität Bonn hat ein ähnliches System namens FotoNav[3] entwickelt. Dort dient das Fotohandy als Fremdenführer. Der Benutzer kann ein Foto von einer Hausfassade machen, dieses per MMS an ihren Server schicken und erhält Informationen über den Standort des Benutzers, beispielsweise in Form eines Kartenausschnitts mit der Position des Benutzers. Für die Bilderkennung wird in diesem System ebenfalls SIFT verwendet.

Weitere Beispiele für ähnliche Systeme finden sich bei Bild[4] und Brigitte[5]. Sendet ein Benutzer dort Fotos eines Artikels ein, erhält er weiterführende Informationen und Videos zum Artikel.

Ebenfalls interessante Literatur zu dem Thema „Mobile Interaktion mit der realen Welt" sind die diesbezüglichen Ausarbeitungen von Gregor Broll im Kontext der Mobile HCI 2008[6] in deren Umfeld auch die Evaluation des Olivanda-Systems präsentiert wurde.

1.3. Struktur der Projektgruppe

Die Projektgruppe war grundsätzlich demokratisch organisiert. Bei Entscheidungen wurde abgestimmt und mit einfacher Mehrheit beschlossen. Eine Ausnahme sind die Vorgaben durch die Betreuer, die immer bindend waren. Diese betrafen vor allem Rahmenbedingungen wie das verwendete Vorgehensmodell XP, den iterativen Abschlussbericht oder die Stundenzettel.

Da das Vorgehensmodell des XP eine Zusammenarbeit mit den Auftraggebern vorsieht, fanden während der gesamten Projektgruppenzeit wöchentlich Treffen mit den Betreuern

[2] http://www.vodafone.de/unternehmen/presse/cebit_pms_128006.html
[3] http://www.ipb.uni-bonn.de/FotoNav/
[4] http://www.bild.de
[5] http://www.brigitte.de
[6] http://www.medien.ifi.lmu.de/team/gregor.broll/

im OFFIS statt. Auf diesen Treffen wurden den Betreuern die in der zurückliegenden Woche durchgeführten Arbeiten präsentiert und gegebenenfalls das weitere Vorgehen besprochen.

In Bezug auf die Implementierung gab es keine festen Verantwortlichkeiten. Jeder konnte jeden Code bearbeiten, solange die Anforderung des Pair Programming eingehalten wurde. Allerdings gab es für besonders anspruchsvolle Themen, wie etwa die Bilderkennung, Experten. Hier war auch der Einarbeitungsaufwand für den Code zu groß, um alle Mitglieder gleichermaßen zu beteiligen.

Zur besseren Koordination der Projektgruppe übte jeder der neun Teilnehmer eine besondere Rolle innerhalb der Gruppe aus. Als Nebeneffekt sollten sich auf diese Weise Experten für bestimmte Themengebiete entwickeln, die im Bedarfsfall als Ansprechpartner dienen konnten, wobei die Verantwortung für diese Teilaufgaben klar definiert war.

Die innerhalb der Projektgruppe vergebenen Rollen und ihre Besetzungen sind in Tabelle 1.1 dargestellt. In den folgenden Unterkapiteln wird jede Rolle von dem oder den zuständingen Gruppenmitgliedern genauer erläutert.

Projektmanagement	Christian Nickel, Oliver Krull
Qualitätsmanagement Software	Oliver Krull
Qualitätsmanagement Usability	Christian Nickel
Softwarearchitekt	Jan Steinke
Hardwarebeauftragter	Jeschua Schang
Entwicklungswerkzeugebeauftragter	Torben Schinke
Event-Management und Öffentlichkeitsarbeit	Marcel Schröder
Webauftritt	Dennis Blum
Corporate-Design-Beauftragter	Sören Samadi
Dokumentenwart	Christian Menke

Tabelle 1.1.: In der Projektgruppe vergebene Rollen und ihre Besetzungen

1.3.1. Projektmanagement

XP adaptieren und umsetzen

Probleme erkennen und bewältigen: Einer der drei Kernaspekte in der Zielsetzung der Projektgruppe war die Verwendung des Vorgehensmodells XP. Dieses Modell basiert in vielen seiner Regeln und Vorgehensweisen jedoch auf der Annahme eines regulären Arbeitsumfeldes mit ganztägiger Beschäftigung zu geregelten Arbeitszeiten und allen Teilnehmern im gleichen Gebäude. Da die Mitglieder der Projektgruppe jedoch während

der Vorlesungszeit einen sehr unterschiedlichen und durchaus vollen Terminplan hatten, musste von einigen empfohlenen oder geforderten Maßnahmen abgerückt bzw. eine Abwandlung dieser Konzepte durchgeführt werden.

Im Zuge dieser Adaption der XP-Techniken hatte das Projektmanagement ein Planungswerkzeug (eXPlain Project Management Tool) auf „Ruby on Rails"-Basis aufgesetzt. Dieses ersetzte zunächst den Backlog und den Iterationsplan, die in klassischer Form (Karteikasten bzw. Whiteboard) nur lokal verfügbar sind und somit für die Gruppe in der Vorlesungszeit unzweckmäßig gewesen wären. Mit diesem Werkzeug war es möglich, Releases und Iterationen zu planen, Storycards und Tasks zu definieren und diese Planungen dem Team über das Internet verfügbar zu machen. Auch die Verteilung der SCs/Tasks war so möglich, ohne das klassische „Stand-Up-Meeting" durchführen zu müssen. (Anmerkung: Die Arbeit mit diesem Tool wurde im Verlauf des Projektes jedoch wieder eingestellt. Die Gründe hierfür werden im Abschnitt 8.5.1 erläutert).

Über die Dauer des Projekts waren noch weitere Probleme bei der Adaption der XP-Dogmen, insbesondere in Bezug auf die Einhaltung der vorgesehenen Zeitrahmen für den Abschluss der Releases und Iterationen, aufgetreten und Teil der Aufgabe des Projektmanagements war es, diese durch möglichst tief gehendes Verständnis von XP vorher zu sehen und bei ihrem Auftreten Lösungen parat zu haben oder sie möglichst zeitnah zu entwickeln. Als Ziel stand hier ein möglichst unfallfreies und schrittweises Annähern an die „harten" XP-Prinzipien, indem das Projektmanagement seine Erfahrungen von Iteration zu Iteration und Release zu Release in die weitere Planung einfließen ließ und auf diesem Wege niemanden aus der Gruppe „verlor".

Zum Einhalten der XP-Regeln motivieren: Damit XP funktionieren kann, sind einige Regeln zu beachten, so beispielsweise das fortwährende Testen mit Unit-Tests und Akzeptanz-Tests, kurze Integrationszyklen und das Programmieren in Zweierteams. Hier lag eine weitere Aufgabe des Projektmanagements darin, die Gruppe dazu zu motivieren, diese Regeln einzuhalten und im Zuge der aus dem Prinzip „Collective Ownership" resultierenden gemeinsamen Verantwortlichkeit, diese Einhaltung zu kontrollieren. Eine zeitnah zu erledigende Aufgabe war in diesem Gebiet das Aufstellen von Gruppenregeln, wie sie von Kent Beck erwähnt werden und die Wahl einer geeigneten Manifestierung dieser Gruppenregeln, wie sie im Abschnitt 1.4 beschrieben wird.

Das Modell XP sieht vor, dass jedes Mitglied gleichgestellt ist und es keinen „Vorgesetzten" gibt. Somit stellte sich als weitere Herausforderung, die Idee von gemeinsamer Verantwortlichkeit und dem „play to win"-Grundsatz in den Köpfen der Teammitglieder zu verankern, damit auch ohne Dyarchie die Einhaltung der Regeln und Planungen gewährleistet war.

Denken, Planen und Walten

Zeitplan ausarbeiten und umsetzen: Ein weiteres Gebiet des Projektmanagements lag in der Ausarbeitung des Zeitplans. Dies umfasste den Release- und Iterationsplan, die

Urlaubsverwaltung und die Lastverteilung über die Semesterphasen. Es wurde bereits zu Beginn vermutet, dass ein konsequentes Verfolgen des XP-Prinzips nur während der vorlesungsfreien Zeit machbar sei. Ein Großteil der Qualitätssicherung entsteht bei XP (unserer gewonnenen Einsicht nach) durch die Gruppendynamik, welche wiederum eine intensive Zusammenarbeit erfordert (oft wechselnde Teams, gemeinsames Refactoring, Stand-Up-Meetings, etc.). Da eine solche Zusammenarbeit nur in der vorlesungsfreien Zeit möglich war, mussten die riskanten und wichtigen Aspekte des Systems in dieser Zeit implementiert werden. Dies wiederum führte zu einer Workload-Ballung. Kent Beck fordert jedoch in seinem Manifest, stets motiviert, ausgeruht und entspannt zu arbeiten. Die Befürchtung war, dass gerade in den Semesterferien diese Motivation nur schwer erlangt/verbreitet werden könnte. Dies hatte sich im Nachhinein nicht bestätigt.

Eine Herausforderung war die langfristige Planung unter dem Umstand der anfangs noch sehr schwammigen Vorstellungen vom endgültigen Erscheinungsbild des zu entwickelnden Systems.

Dokumentieren: Dem „play to win"-Grundsatz folgend sollte das Projektmanagement sein Möglichstes dazu beitragen, die Gruppennote so gut wie möglich ausfallen zu lassen. Daher war es Teil seiner Aufgabe, die Gruppe dahingehend zu motivieren, ihre Arbeit, wie von den Betreuern gefordert, zu dokumentieren. Auf diese Weise sollte es möglich sein, einer Workload-Ballung zum Ende des Projekts aus dem Weg zu gehen. Erfahrungsgemäß führt die Doppelbelastung von letzten Änderungen und Verfassen des Endberichts zu Motivationseinbrüchen, obwohl gerade jener Bericht einen wohl nicht unwesentlichen Teil der Bewertung ausmachte.

Und das alles...

... bei Aufrechterhaltung des guten Gruppenklimas: Ebenso war es ein Ziel des Projektmanagements, das Gruppenklima so gut wie möglich zu gestalten. In diesem Gebiet zeichneten sich jedoch (im Gegensatz zu den anderen Gebieten) keinerlei Probleme ab. Es wurde erreicht, dass die Arbeitslast sowohl über die Zeit, als auch innerhalb der Gruppe so verteilt war, dass sich niemand ungerecht behandelt fühlte und es wurde dadurch vermeiden, dass es zu einer gereizten Stimmung kam. Im Rahmen der kollektiven Verantwortlichkeit wurde stets versucht, dieses Ziel innerhalb der Gruppe zu propagieren.

1.3.2. Qualitätsmanagement Software

Rollenverständnis

Der Qualitätsmanager Software achtete darauf, dass der Quellcode den Codekonventionen folgt und wies gegebenenfalls Gruppenmitglieder, die diese Konventionen nicht einhielten, darauf hin, dies fortan zu tun. Auch achtete er darauf, dass der Quelltext sinnvoll

kommentiert wurde und JavaDoc-Kommentare eingesetzt wurden, um die Verständlichkeit und Übersichtlicheit des Codes zu erhöhen.

Weiterhin stand er bei Fragen zur Strukturierung des gesamten Projektes als Ansprechpartner zu Verfügung. Besonders beim Refactoren und Integrieren einzelner Teilpakete im Projekt achtete er darauf, dass diese Integration so verlief, dass die Codequalität gewahrt blieb und keine „zusammengestrickten Workarounds“ entstanden. Stark zur Codequalität beitragen sollten die Technologien „JUnit“ und „JMUnit“ (im weiteren Verlauf des Projektes verworfen), deren Einsatz der Qualitätsmanager Software überwachte.

Durchgeführte Tätigkeiten zu Beginn des Projektes

Zu Beginn der Implementierungsphase wurden Gruppenmitglieder bereits darauf hingewiesen, dass die Java-Codekonventionen möglichst strikt eingehalten werden, um ein einheitliches Bild des Quellcodes zu erreichen.

Weiterhin wurde die Teilung des Projektes in zwei Teilprojekte angeregt, wobei ein Teilprojekt die Implementierung auf Seiten des Mobiltelefons beinhaltet und das andere Teilprojekt die Implementierung des Servers einschließt.

Weitere Tätigkeiten

Auch im weiteren Verlauf achtete der Qualitätsmanager Software stark darauf, dass alle Mitglieder die Codekonventionen einhielten und damit eine sauber geschriebene Software entstand, die auch eventuelle externe Programmierer verstehen können.

1.3.3. Softwarearchitekt

Der Aufgabe des Softwarearchitekten entspricht im XP der Rolle des „Architekten“. Daher fielen in seinen Aufgabenbereich vor allem die Beaufsichtigung der im Laufe des Entwicklungsprozesses fortlaufend durchzuführenden Refactorings. Hierzu war es nötig, immer einen Gesamtüberblick des Systems zu besitzen, um jederzeit über eventuell vorhandene Schwachstellen in der Architektur im Bilde zu sein und die Projektgruppenmitglieder darauf gegebenenfalls hinweisen zu können. Desweiteren musste der Softwarearchitekt darauf achten, dass Refactorings auch tatsächlich durchgeführt wurden.
Durch die Verwendung des Vorgehensmodells XP und des damit verbundenen „Pair Programmings“ wurde erreicht, dass sich jeder mit jedem Teil des Systems zumindest oberflächlich auskannte. Nicht jedes Paar im „Pair Programming“ konnte jederzeit an allen Stellen des Systems arbeiten, daher war der Softwarearchitekt ein Ansprechpartner für Fragen der jeweils aktuellen Softwarearchitektur und musste gegebenenfalls Fragen an die momentan in dem jeweiligen Bereich tätigen Programmierer delegieren können.

Zusätzlich: Der Integration-Designer

Die Aufgabe des Integration-Designers war im Projekt Olivanda nicht vergeben und es wurden daher gemeinsam unter Mitwirkung durch den Kunden (die Betreuer) Story-

cards entwickelt. Nach Festlegung der Storycards für jede Iteration war es Aufgabe des Teams, Tasks für die einzelnen SCs zu erstellen.
Da es zu den Zielen des Softwarearchitekten zählte, immer den Überblick über das Gesamtsystem zu besitzen, hatte er die zusätzliche Aufgabe, die zu jeder Storycard gefundenen Tasks fortlaufend sowohl auf Sinnhaftigkeit, als auch auf einen ausreichenden Detailgrad zu überprüfen. Auf diese Weise trug er dazu bei, die Gestaltung der Storycards und Tasks im Laufe des Entwicklungsprozesses der XP-Adaption zu verbessern. Durch eine optimale Definition von Tasks wurde es erleichtert, eine gute Softwarearchitektur zu erstellen und zu pflegen.

1.3.4. Hardware-Beauftragter

Verständnis

Der Hardwarebeauftragte war für alle Fragen zuständig, die die Hardware betrafen. Hardware war zum einen das mobile Gerät, aber auch der stationäre Backend-Server. Wann immer in der Projektgruppe Fragen bezüglich der Hardware auftauchten, sollten sie (wenn sinnvoll) dem Hardwarebeauftragten gestellt werden. Dieser kümmerte sich dann um die Lösung des Problems.
Der Hardwarebeauftragte sollte stets den Überblick über die in der PG eingesetzten Hardware haben. Das heißt, welche Geräte sind aktuell in der Architektur eingebunden, welche Geräte sind nicht eingebunden? Welcher Teilnehmer hat aktuell welches mobile Gerät?
Weiterhin sollte der Hardwarebeauftragte in gewissem Rahmen wissen, welche zur Verfügung stehenden Geräte welche Funktionalität boten oder nicht boten.
Der Hardwarebeauftragte sollte weiterhin eine gute Kenntnis der in den Mobilgeräten verfügbaren API haben und für Fragen bzgl. der API zur Verfügung zu stehen. Er sollte sich einigermaßen in die API einarbeiten und wissen, wie man auf bestimmte Funktionen des mobilen Gerätes zugreift.

Zielsetzung

Die Zielsetzungen umfassten im Prinzip die oben genannten Aufgaben:

- Kenntnis der eingesetzten Hardware haben
  - Funktionalität kennen
  - Fragen darüber beantworten können
- Stets den Überblick über die Hardware in der PG haben
- Tiefergehende Kenntnis der APIs der verwendeten Programmiersprachen.

Was wurde realisiert

Die der PG zur Verfügung stehenden mobilen Geräte (zwei Nokia N95 Mobiltelefone) wurden den Teilnehmern vorgestellt. Anschließend erfolgte eine weitere Einarbeitung in die Geräte und ihre Funktionalität. Die Programmierung für die mobilen Geräte sollte dem S60 Standard entsprechen, weshalb dort ebenfalls eine Einarbeitung in die API stattfand und der PG im Rahmen erster Tests mit GUI-Programmierung ein Einblick gegeben wurde.
Im weiteren Verlauf der Entwicklung fand eine Umstellung auf das J2ME-Polish Toolkit statt. Dort gab es ebenfalls eine Einarbeitung in die API, speziell auch auf die Möglichkeiten der GUI-Programmierung mit J2ME-Polish hin. Dort stand der Hardwarebeauftragte dann für Fragen bzgl. der API zur Verfügung.
Serverseitig wurde das Frontend zum Verwalten von Kampagnen, Diensten und Plakaten entwickelt. Dort stand der Hardwarebeauftragte ebenfalls für Fragen bzgl. der PHP-API oder Datenbankfragen zur Verfügung.

1.3.5. Entwicklungswerkzeuge-Beauftragter

Im XP gibt es keine definierte Rolle eines Verantwortlichen für Entwicklungswerkzeuge. Daher wird zunächst eine Definition von Entwicklungswerkzeugen gegeben. Entwicklungswerkzeuge sind Programme, die von Programmierern benutzt werden, um Anwendungen zu erstellen. Diese Programme werden häufig in integrierten Entwicklungsumgebungen aggregiert. Dazu zählen beispielsweise Compiler, Debugger oder Eingabemöglichkeiten für Quellcode. Weiterhin zählen externe Programme wie Build-Tools, Quelltext-Generatoren und Testsoftware dazu. Schwierigkeiten bereitet hier die Abgrenzung zu Frameworks wie JUnit, Axis oder Ergänzungsbibliotheken wie J2ME Polish.

Überblick der Werkzeuge

Jeder in der Projektgruppe hatte die Rolle eines Programmierers inne, daher bildete die Kenntnis über die Entwicklungswerkzeuge eine wesentliche Basis zur Steigerung der Effizienz und der Qualität der Software. Es kann sich jedoch nicht jeder mit diesen Werkzeugen auskennen, daher stand der Entwicklungswerkzeuge-Beauftragte für Anfragen zur Verfügung und konnte Werkzeuge und Frameworks in ihrer Benutzbarkeit und Leistungsfähigkeit wenn nötig erklären.

Verwendung von Werkzeugen

Es wurden Java 6 für den Server und MIDlets für die mobilen Endgeräte mit dem J2ME Profil MIDP 2.0 verwendet. Die Entwicklung spezialisierte sich auf die Endgeräte mit Symbian S60 Plattform. Netbeans bot zur unabhängigen Projekterstellung die automatische Generierung von Ant-Files. Serverseitig fand eine testgetriebene Entwicklung mit

JUnit 4 statt. Clientseitig war dies nicht überall einsetzbar, da dort vorwiegend API-Aufrufe stattfinden. Tests wären dort auch sehr aufwändig, da diese auf dem mobilen Gerät ausgeführt werden müssten. Zur Unterstützung der Clientprogrammierung wurde das Framework J2ME-Polish verwendet.

1.3.6. Event-Management und Öffentlichkeitsarbeit

Rollenverständnis

Eventmangement und Öffentlichkeitsarbeit war als Schnittstelle zwischen der Projektgruppe bzw. dem Projekt und der Öffentlichkeit anzusehen. Die Rolle betraf die Organisation von Veranstaltungen wie einem Fußballturnier oder ähnlichen Events, die dazu da sein sollten, das Gruppengefühl zu stärken. Weiterhin sollte das einheitliche Auftreten der Projektgruppe in der Öffentlichkeit gefördert werden.
Aufgabe des Beauftragten für Event-Management und Öffentlichkeitsarbeit war die vernünftige Repräsentation des Projekts auf der Homepage. Diese sollte stets den aktuellen Stand des Projekts anzeigen.
Außerdem war die Rolle als kontrollierende Instanz bezüglich allen veröffentlichten Dokumenten zu sehen. So sollten beispielsweise öffentliche Dokumente in Bezug auf Corporate Design einheitlich sein und inhaltlich dem Projekt entsprechen.

Realisierung

- Organisatorische Tätigkeiten für das Blockseminar
- Organisation der Weihnachtsfeier
- Ausflug nach Bremen, um die Projektgruppe bei Ströer zu repräsentieren
- Organisation von diversen Grillfeiern und Fußballtrainings
- Kontaktaufname mit Herrn Brosius von Vodafone für Recherchen
- Mitarbeit am Frontend und insbesondere am Annotaionsfrontend des Handys
- Einführung von News auf der Homepage
- Erstellung eines Beispielvideos auf der Homepage

1.3.7. Webauftritt

Rollenverständnis

Der Webseiten-Beauftragte war verantwortlich für das Einrichten, Erstellen und Pflegen der Homepage. Das Layout/Design der Homepage wurde mit dem Corporate-Design-Beauftragten abgesprochen und anschließend entsprechend umgesetzt. Desweiteren war

der Webseiten-Beauftragte Ansprechpartner für die Projektgruppe, wenn es um gewünschte Inhalte oder Änderungen der Homepage ging.

Realisierung

Die Homepage wurde unter folgender Adresse eingerichtet:

http://srvmi01.offis.uni-oldenburg.de/olivanda/

Damit die Inhalte einfach und schnell erstellt werden konnten, wurde das Content-Management-System Drupal eingerichtet. Die Drupal-Templates ließen sich relativ einfach ändern, so dass das Design der Homepage den Corporate Design Anforderungen gut angepasst werden konnte.

Der Inhalt der Homepage gliederte sich in folgende Punkte:

- Startseite
- News
- Aktueller Projektfortschritt
- Projektbeschreibung
- Evaluation des Verfahrens
- Evaluation der Dienste
- Bilder
- Seminarabeiten
- Kontakt
- Links

Auf der Startseite wurde das Projekt kurz vorgestellt, wobei das Hauptaugenmerk auf der Motivation und Zielsetzung lag. An der rechten Seite konnte man zusätzlich ein Video von unserem aktuellen Client sehen.
Die News-Seite beinhaltete immer die aktuellen Neuigkeiten und Ankündigungen.
Unter dem Punkt *Aktueller Projektfortschritt* gab es einen Fortschrittsbalken mit den einzelnen Releases und Iterationen. Hier konnte der Projektfortschritt wochengenau mitverfolgt werden.
Unter Projektbeschreibung war die ausführliche Projektbeschreibung zu finden.
Die Seiten *Evaluation der Interaktionsmöglichkeiten der Anwendung* und *Evaluation der Dienste* beschrieben unsere beiden Evaluation und die Ergebnisse. Bei der zweiten Evaluation entstand ein Video, dass ebenfalls auf der Seite zu sehen war.

Unter *Bilder* waren vier Alben zu finden. Im ersten Album waren die Bilder des Seminarwochenendes an der Thülsfelder Talsperre, im zweiten und dritten Album waren Bilder von der ersten Evaluation und deren Vorbereitung und im vierten Album waren Bilder von der PG beim arbeiten.
Die Seite *Seminararbeiten* enthielt von jeder in in der Seminarphase entstandenen Seminarabeiten eine kurze Zusammenfassung. Die vollständigen Arbeiten und die Präsentationen wurden unter zum download als PDF bereitgestellt.
Auf der Kontakt-Seite ist unser Ansprechpartner mit Telefonnummer und die Email-Adresse zu unserem Verteiler angegeben.
Unter Links konnte man einige nützlich Links finden.

1.3.8. Corporate-Design-Beauftragter

Rollenverständnis

Das Ziel des Verantwortlichen für Corporate-Design in der Projektgruppe bestand darin, durch visuelle Gestaltung zur Corporate Identity der Projektgruppe beizutragen. Hierbei war es wichtig, einerseits eine Akzeptanz seitens der Projektteilnehmer zu erreichen und andererseits einen hohen Wiedererkennungswert bei Stakeholdern und Interessenten zu erreichen. Die Corporate Identity half den Projektgruppenmitgliedern, sich mit selbiger zu identifizieren, was in einer erhöhten Produktivität mündete, da sich die Mitglieder so eher involviert fühlten.

Zum Corporate Design gehörten unter anderem die Gestaltung der Kommunikationsmittel wie Logos, Werbemittel, Verpackungen, Folienlayouts und Gestaltung der Onlinemedien.

Realisierung

Im Rahmen der Tätigkeit des Corporate Designers wurde ein Logo entworfen, das der Projektgruppe als Identifikationspunkt diente und den Wiedererkennungswert auf den gegebenen Unterlagen steigern sollte. Weiterhin wurde ein einheitliches Folienlayout für Präsentationen entworfen, um die Uniformität der Dokumente zu erhalten.

Das Logo wurde den üblichen Design-Prinzipien unterworfen, welche folgende Richtlinien abdecken:

- Verständlichkeit, das bedeutet eine Unterstreichung der Bedeutung oder ein Hinweis auf die Tätigkeit der Projektgruppe.
- Unverwechselbarkeit, denn ein Logo transportiert das Image der Projektgruppe. Abgesehen von rechtlichen Problemen welche aus unzureichender Unverwechselbarkeit entstehen, behindern Fremdassoziationen mit anderen, ähnlichen Logos die Identifizierung der Projektgruppe mit dem Logo.



Abbildung 1.1.: Olivanda Logo v1

- Einprägsamkeit. Es gilt die Regel KISS : „Keep It Short (and) Simple“ – was einfach ist, ist einfach zu merken.
- Reproduzierbarkeit. Ein Logo muss leicht nachzuzeichnen sein und auch bei niedrigen Auflösungen noch klar erkennbar sein. Farbenfrohe Logos sind zwar technisch kein Problem, jedoch muss das Bild auch auf Logolabels von Produkten gut erkennbar sein, aber auch als Fax, als Stempel, aus der Entfernung, auf einem T-Shirt gestickt oder auf dem Werbekugelschreiber noch gut aussehen.
- Anforderungen. Ein Logo enthält Hauptelemente und Nebenelemente. Hauptelemente sind der Firmenname und die angebotene Leistung, oder das angebotene Produkt. Ggf. verzichtbare Nebenelemente sind grafische Elemente, ein Slogan, oder eine Identitätsaussage.

Das Logo wurde demnach bewusst einfach gehalten, um den Punkten Reproduzierbarkeit und Einprägsamkeit zu entsprechen. Als Anforderungen wurde bestimmt, dass der Name der Projektgruppe und eventuell die entwickelnde Abteilung visualisiert wird. Um die Verständlichkeit zu unterstützen, wurde ein Kreisförmiger Ansatz gewählt, um einerseits eine grafische Brücke zwischen Bild und Text zu bilden, andererseits um metaphorisch die Übertragung zu skizzieren. Die weiteren Kreise sollen Funkwellen darstellen, welche die Übertragung simulieren. Weiterhin weist ein Kreis auf eine „globalere“ Anwendung hin, da auch eine Art Globus hineininterpretiert werden kann. Die Unverwechselbarkeit wird dadurch gewährleistet, dass das Logo der Projektgruppe nicht an anderen Visualisierungen angelehnt ist.

Nach Feedback durch die Gruppe wurde eine Revision des Logos angeregt. Kritikpunkte waren unter anderem:

- Zu konstrastarme Farbwahl
- Nicht zeitgemäßes Design
- Durch Assoziierung mit dem ehemaligen ZDF-Logo wurde der Punkt der Unverwechselbarkeit nicht erfüllt

Daher fand eine Neugestaltung des Logos auf Basis der gegebenen Kritik statt. Um ein moderneres Erscheinungsbild zu gewährleisten, wurden Glas-Effekte und Reflexionen be-

nutzt um das Logo auffälliger und ansprechender zu gestalten. Zusätzlich fanden hellere Farben Verwendung, die sich auch auf der Homepage der Projektgruppe wiederfinden und so ein harmonischeres Gesamtbild erzeugen. Das Kreisprinzip aus dem ersten Logo wurde aufgegriffen und verfeinert, indem eine Kameralinse mitsamt Sucher simuliert wurde, was auf den vorrangigen Verwendungszweck der Software hindeuten sollte. Das Feedback der Gruppe zu dem neuen Logo (vgl. 1.2)war überwiegend positiv.



Abbildung 1.2.: Olivanda Logo v2

Folgend fanden mit Hilfe weiterer Kritik und Anregungen weitere Feinabstimmungen statt, bis schließlich eine allgemeine Zustimmung zu dem Logo erreicht wurde. Grundgedanke der weiteren Revisionen war, eine Iris (vgl. Abbildung 1.4) statt der Kamera als Metapher des „Fokus “zu nutzen. Schließlich wurde aber eine Kombination aus beidem als gelungenste Fassung (vgl. Abbildung 1.5) auserkoren.



Abbildung 1.3.: Olivanda Logo v2



Abbildung 1.4.: Olivanda Logo v2.2



Abbildung 1.5.: Olivanda Logo Final

Weitere Tätigkeiten

Das weitere Auftreten der Projektgruppe wurde sowohl von der Farbwahl als auch von den Formen sämtlicher grafischer Elemente angepasst, um ein geschlossenes Bild zu vermitteln. Als markante Farbe der Projektgruppe wurde ein helles Himmelblau (Farbcode: #a0c7ec) genutzt.

Das Logo war als wiederkehrendes Element auf vielfältige Weise genutzt worden, so zum Beispiel als Header-Grafik im Frontend oder im Hintergrund des Handyclients. Das übrige Layout setzte sich zumeist aus eckigen Formen und modernen, glasartigen Strukturen zusammen, welche in helleren Blautönen gehalten waren.

Obwohl zur Nutzung des entwickelten Corporate Designs zwar kein Corporate-Design-Handbuch beziehungsweise kein Leitfaden zur Anwendung entworfen wurde, war es dennoch möglich anhand der visuellen Basiselemente das Design weiterzuentwickeln und gut einzubinden. Die Kombination aus Wiedererkennbarkeit und praktischem Nutzem war durchweg gelungen, da der Großteil der Entwürfe sowohl bei Projektgruppen-Mitgliedern als auch bei Betreuern positiv entgegengenommen und als optisch ansprechend empfunden wurden.

1.3.9. Dokumentenwart

Der Dokumentenwart war der Ansprechpartner für alle Gruppenmitglieder, wenn es um das Archivieren von Dokumenten oder das Wiederauffinden archivierter Dokumente ging. Er behielt den Überblick über alle im Verlauf des Projekts entstandenen Dokumente und insbesondere den Abschlussbericht.

Er war auch dafür verantwortlich, dass Routinearbeiten wie das Erstellen von Sitzungsprotokollen durchgeführt wurden und dass diese eher lästigen Arbeiten gleichmäßig auf die Gruppenmitglieder verteilt wurden. Mit Hilfe von Vorgaben und Vorlagen sorgte er – in Absprache mit dem Corporate Design-Beauftragten – dafür, dass solche Dokumente ein einheitliches Aussehen bekamen.

Bei der Erstellung des Projektgruppenberichts war er für das Zusammenführen der Inhalte zuständig, kümmerte sich um die technischen Rahmenbedingungen des Berichts

und sorgte für ein einheitliches Aussehen des Textsatzes. Er konnte auch Teilaufgaben bei der Bearbeitung bzw. Erstellung von Texten an die Gruppenmitglieder delegieren.

Zu guter Letzt war er Ansprechpartner, wenn Probleme bei der Erstellung von Dokumenten auftraten, etwa bei der Arbeit mit LaTeX, oder Unklarheiten beispielsweise beim Stil von Quellenangaben.

Durchgeführte Tätigkeiten

Während der Seminarphase des Projekts stand der Dokumentenwart als Ansprechpartner bei Problemen mit dem Textsatz zur Verfügung. Da mit dem verwendeten ACM-Template Probleme auftraten, wurden tatsächlich einige Recherchearbeiten nötig.
Im weiteren Verlauf der Projektgruppe wurde vom Dokumentenwart eine Vorlage für die Sitzungsprotokolle erstellt. Nach Beschluss der Projektgruppe wurden die Protokolle im SVN archiviert, so dass der Aufwand für diese Arbeit de facto entfiel – stattdessen mussten die Protokolle durchgesehen und die Protokollanten ggf. um Nachbesserungen gebeten werden.

Bei der Erstellung des Endberichts koordinierte der Dokumentenwart die anfallenden Tätigkeiten und erstellte regelmäßig zur Dokumentationswoche Aufgabenlisten sowie TODO-Hinweise im Bericht.
Zur Entlastung des Managements erstellte er die von den Betreuern geforderten Retrospektiven inklusive der Bilanzierung der Aufwandsabschätzungen, die in den Besprechungen präsentiert und im Bericht zusammengefasst wurden. Für die Retrospektive wurden die Gruppenmitglieder regelmäßig zum Iterationsende aufgefordert, an einem Brainstorming im Forum teilzunehmen.
Als gegen Ende des Projekts die Dokumentationswoche für Programmierarbeiten bzw. Bugfixing benötigt wurde, übernahm der Dokumentenwart den Dokumentationsaufwand größtenteils selbst, um anderen Gruppenmitgliedern Zeit für die Umsetzungsarbeiten zu verschaffen. In den letzten Wochen wurde von ihm schließlich die Überarbeitung und Korrektur des Endberichts in ausgedruckter Form organisiert.

1.4. Gruppenregeln

Zur Vereinfachung der Zusammenarbeit unter den Gruppenmitgliedern existierten feste Regeln, die von allen Gruppenmitgliedern einzuhalten waren. Diese bezogen sich im Wesentlichen, aber nicht ausschließlich auf Programmierarbeiten.

- Benutzung von JUnit 4:
  JUnit ist ein Framework zur Unterstützung von Unit-Tests und automatisierten Ausführung von Tests. Die Idee dahinter liegt in der testgetriebenen Entwicklung – erst Testen, dann implementieren. Die Testfälle sollen nach jeder Programmänderung durchgeführt werden und jeder Testfall testet nur eine bestimmte Funktion.

- Javadoc-Kommentare:
  Javadoc ist ein Software-Dokumentationswerkzeug, das aus Java-Quelltexten automatisch HTML-Dokumentationsdateien erstellt. Zur besseren Einsicht und für eine mögliche spätere Fortführung des Projekts wurden für alle Funktionen Javadoc-Kommentare angelegt.
- Kommentare auf Englisch:
  Die Kommentare wurden englischsprachig gehalten, um den Code international verständlich zu halten.
- Methodennamen auf Englisch:
  Die Methoden wurden ebenfalls englischsprachig gehalten, um den Code international verständlich zu halten.
- Runtime-Exceptions werfen, wenn eine Vorbedingung nicht eingehalten wird. Keine Nullwerte im Fehlerfall liefern.
- Ausgaben über Log4J:
  Alle Status- und Fehlermeldungen auf der Serverseite sollten über das Log4J-Framework ausgegeben werden. Die Konsole sollte nicht direkt benutzt werden. Auf diese Weise konnten alle Logging-Ausgaben zentral (de-)aktiviert und umgeleitet werden.
- toString()-Methode einer jeden Klasse überschreiben:
  Da das benutzte Logging-Framework sich zur Beschreibung von Log-Event-Quellen der toString()-Methode bedient, sollten diese Methoden in allen Klassen überschrieben werden. So wurden die Logging-Ausgaben leichter lesbar, da man sinnvolle Beschreibungen statt JVM-Objekt-IDs als Quellen zurückbekam.
- Es durfte auch alleine dokumentiert werden:
  Um möglichst effektiv zu arbeiten, konnten Dokumentationsarbeiten auch alleine durchgeführt werden – der für das Programmieren wichtige Vorteil, dass der entstehende Code weniger Bugs enthält, war hier nicht relevant. Der Verzicht auf Arbeit in Zweierteams ermöglichte eine flexiblere Planung beziehungsweise Heimarbeit.
- Testvorbereitungen und -nachbereitungen durften auch alleine durchgeführt werden:
  Die im Rahmen der Testgetriebenen Entwicklung nötigen Tests durften auch ohne Pair vorbereitet werden, um so beim eigentlichen Implementierungstreffen schneller voran zu kommen und mögliche Probleme bereits vorab festzustellen.
- Geldstrafen für Verspätungen an den Gruppentreffen mit den Betreuern:
  - 1-4 Minuten: 10 Cent pro Minute
  - ab 5 Minuten: 50 Cent pro Minute
  - Höchstsatz: 2 Euro

1.5. Kooperation mit Ströer Out-of-Home Media

Im Rahmen des OFFIS-Projekts Intermedia ergab sich die Möglichkeit zur Zusammenarbeit mit dem Werbeunternehmen Ströer Out-of-Home Media. Da Ströer Media vor allem im Bereich der Plakatwerbung tätig ist, konzentrierte sich die Projektgruppe insbesondere auf dieses Medium. Die Kooperation hatte auch Auswirkungen auf die einsetzbaren Methoden zur Erkennung von Medien. So durften z.B. keine Modifikationen an den Plakaten nötig werden, um die Erkenung zu ermöglichen.

Im Zusammenhang mit dieser Kooperation bot die Universität Oldenburg im Sommersemester 2008 eine Lehrveranstaltung „Konzeption, Realisation und Evaluation einer Out-of-Home-Medienkampagne mit echten Kunden" an, an der zwei Mitglieder der Projektgruppe teilnahmen. Aus dieser Veranstaltung stammten auch die in der zweiten Evaluation verwendeten Plakate.

1.6. Globale Zeitplanung

Die Zeitplanung für das gesamte Projekt ließ sich, bedingt durch das verwendete Vorgehensmodell XP, in vier große Releases einteilen, die sich jeweils in einzelne Iterationen weiter unterteilen ließen. In Abb. 1.6 wird dargestellt, in welcher Kalenderwoche sich eine Iteration und dessen Release befindet. Der jeweils aktuelle Stand innerhalb des Zeitplanes wird im Laufe des Endberichtes zu Beginn jedes Releases aktualisiert dargestellt.
Die unregelmäßige Dauer der einzelnen Releases bzw. Iterationen kamen durch kurzfristige Zeitplanänderungen während des Projektes zustande. So hatte sich beispielsweise das zweite Release durch eine Evaluation des Systems oder das Verschieben von Treffen in Folge von anderen Vorlesungen verlängert.

Kalenderwoche	Release	Iteration	Woche
(01.01.2008)	1	1	1
3	1	2	1
4	1	3	1
5	1	4	1
(04.02.2008)	2	1	1
7	2	1	2
8	2	1	3
9	2	2	1
10	2	2	2
11	2	2	3
12	2	2	4
13	2	3	1
14	2	3	2
15	2	3	3
16	2	3	4
17	2	3	5
18	2	3	6
19	2	3	7
(12.05.2008)	3	1	1
21	3	1	2
22	3	1	3
23	3	1	4
24	3	2	1
25	3	2	2
26	3	2	3
27	3	2	4
28	3	3	1
29	3	3	2
30	3	3	3
31	3	3	4
(04.08.2008)	4	1	1
33	4	1	2
34	4	1	3
35	4	1	4
36	4	2	1
37	4	2	2
38	4	2	3
(30.09.2008)	4	2	4

Abbildung 1.6.: Übersicht der globalen Zeitplanung

2. Zielsetzung

Die Festlegung sowohl allgemein gefasster Leitmotive als auch konkreter, in Software umzusetzender Ziele erfolgt in diesem Kapitel. Die eigentlichen Ansätze zur Verwirklichung dieser Ziele sind dabei zunächst von untergeordneter Bedeutung; sie werden im Kapitel 3 detailliert beschrieben.

2.1. Leitszenario

Marcel Schröder hat folgendes Leitszenario für die Projektgruppe verfasst:

> Die reale Welt sollte benutzerfreundlich mit verschiedenen „Sinnen“ eines mobilen Endgeräts erfasst werden, um sie mit digitalen, aufbereiteten Informationen sinnvoll zu ergänzen. Weiterhin sollte das „Gedächtnis“ des Endgerätes sich die Informationen ggf. merken und assoziieren. Dabei stand die Erkennung und Informationsergänzung von Plakaten im Vordergrund.

2.2. Konkrete Ziele

Entsprechend des vorgegebenen Rahmens und der groben Zielsetzung durch das Leitszenario wurde – auch im Hinblick auf die Zusammenarbeit mit Ströer Out-of-Home Media – zunächst ein einzelnes konkretes Ziel gesetzt, das mit Hilfe des agilen Softwareentwicklungsprozesses erreicht werden sollte.

Nach Verwirklichung dieses ersten Teilaspekts wurde die Aufgabenstellung um eine Kontextbezogenheit der gelieferten Informationen erweitert, um ihren Nutzen zu steigern.

2.2.1. Informationsergänzung bei Plakaten

Das erste und wichtigste Ziel war die Informationsergänzung bei Plakaten. Diese sollte mit Hilfe von mobilen Endgeräten mit eingebauter Kamera durchgeführt werden; die genaue Vorgehensweise wird in Kapitel 3 beschrieben. Insbesondere durfte die Plakaterkennung keine zusätzlichen Anforderungen an das Erscheinungsbild oder die Herstellung der Plakate stellen.

2.2.2. Kontextsensitivität der Informationsergänzung

Zweites Ziel war die kontextabhängige Modifikation der durch die Plakaterkennung gelieferten Informationen. So sollten beispielsweise die aktuelle Uhrzeit bzw. das Datum und auch der Aufenthaltsort des Benutzers Einfluss darauf haben, welche Informationen zu einem Plakat bereitgestellt werden. Durch diese Maßnahme sollte verhindert werden, dass der Benutzer mit für ihn nicht relevanten Informationen überflutet wird und es sollte die Qualität der gelieferten Informationen gesteigert werden.

2.2.3. Annotation unbekannter Objekte

Ein weiteres Ziel war die Möglichkeit, aufgenommene Plakate, die vom Server nicht erkannt wurden, mit Hilfe des Mobiltelefons hinzuzufügen. Dazu kann der Benutzer über den im Handy integrierten Browser eine Kampagne mit dem soeben aufgenommenen Foto ergänzen, oder aber eine neue Kampagne anlegen. Das Design der Oberfläche des Browser-Frontends auf dem Handy sollte dabei identisch zum Design der Oberfläche des regulären Web-Frontends sein. Durch die Annotation über das Handy sollte es auch möglich sein, zu einem bereits vorhandenen Plakat verschiedene Ansichten hinzuzufügen. So könnte eine Kampagne, die die Ansicht eines Stuhls enthält, um weitere Ansichten dieses Stuhls ergänzt werden.

Durch diese Annotations-Möglichkeiten wird der Nutzer direkt in die Erweiterung des Plakatbestandes im Sinne von „User-generated content“ einbezogen.

2.3. Umsetzung des Leitszenarios

Um die korrekte Umsetzung des Leitszenarios bei der Abnahme zu verifizieren wurde eine Demonstration anhand eines Szenarios geplant. Dieses Szenario bestand aus einer Teilmenge von vorher erstellten Szenarios um in dem gegebenen Zeitrahmen ein Maximum an Funktionen abzudecken. Das genutzte Basis-Szenario bestand aus der Annotation eines Mensaflyers. Die Szenariobeschreibung lautete wie folgt:

> Peter Beispiel möchte eine öffentliche Feier veranstalten. Dazu druckt er Flyer und verteilt sie in der Mensa. Für Außenplakate in den Uni-Werbeboxen ist sein Kumpel Andi verantwortlich, der daher als registrierter OLIVANDA Nutzer über sein Handy die Kampagne mit diesem Plakat erweitert. Ziel der Kampagne ist es weiterführende Informationen an potentielle Nutzer zu übermitteln. Marie Musterfrau, als nicht registrierte Nutzerin, entdeckt beim Mensen den Flyer und wählt einen Dienst. Klaus Klausen, als registrierter Nutzer, fotographiert ein Plakat aus der Werbebox und wählt einen Dienst. Danni Dussel findet den Flyer zwei Wochen zu spät.

Der Ablauf des Szenarios fand wie folgt statt. Ein Projektgruppenmitglied, welches die Person „Peter“ spielte, meldete sich am OLIVANDA-Frontend an und erstellte eine neue Kampagne. Dazu hatte er seinen Mensaflyer hochgeladen. Er erstellte weiterhin folgende Dienste:

- Zeige mir eine mögliche Anfahrt (singelton: drive, rank:: typ=bike)
- Zeige mir eine mögliche Anfahrt (singelton: drive, rank:: typ=car, location=all)
- Zeige Musikrichtung 20-22 Uhr (rank:: music=pop)
- Zeige Musikrichtung 22-24 Uhr (rank:: music=rock)
- Zeige Musikrichtung 24-02 Uhr (rank:: music=metal)
- Ein Restaurant in der Nähe (rank:: food=mexican)
- Ein Restaurant in der Nähe (rank:: food=italian)
- Sende e-Mail „Bock auf nächste Party“ an Peter nach der Feier-Deadline

Ein zweites Gruppenmitglied spielte die Person Andi, welche im Szenario bereits registriert war und mit OLIVANDA ein Foto von dem Plakat machte. Dazu gab er zuerst seine Nutzerdaten am Handy ein und wählte, nachdem das Plakat korrekterweise nicht erkannt wurde, die Option zum Hinzufügen an eine vorhande Kampagne. Als Kampagne benutzte er die von Peter erstellte. Das dritte Gruppenmitglied, „Marie“, entdeckte den Flyer von Peter. Über ein mit dem Handy geschossenes Foto erhielt es einen Anfahrtsdienst, drei Musiklisten in irgendeiner Reihenfolge und zwei Restaurants in der Nähe in irgendeiner Reihenfolge. Die Anfahrtssuche verwendete die zuletzt bekannten GPS Koordinaten.

Das letzte Mitglied („Klaus“) registrierte sich zunächst als Nutzer bei OLIVANDA und gab sein Profil an (Mobilität, Essensvorlieben, Musikgeschmack, ...). Danach bemerkte er Peters Plakat auf dem Campus. Er machte ein Photo und erhielt eine Anfahrtsskizze sowie sortierte Listen der Musik und Restaurants passend zu seinem Profil. Schließlich fand ein Gruppenmitglied im Szenario den Flyer 2 Wochen zu spät und erhielt als Dienst nur noch eine Seite, auf der er eine E-Mail an Peter schicken konnte, dass er wieder eine weitere Feier veranstalten solle.

3. Konzeptuelle Lösung

In diesem Kapitel werden die Methoden zum Erreichen des in Kapitel 2.2 festgelegten konkreten Ziels recherchiert und evaluiert. Am Ende des Kapitels soll eine grobe Roadmap stehen, welche Verfahren die in der Projektgruppe entstandene Software einsetzt und wie diese miteinander zu verknüpfen sind.

3.1. Mobile Anwendung

Die eigentliche Bildakquisition sollte durch gängige Mobiltelefone stattfinden, auf denen zu diesem Zweck ein Client installiert werden muss. Er übernimmt auch das Bereitstellen der Zusatzinformationen bzw. Dienstleistungen zu den fotografierten Plakaten, wobei aufgrund der beschränkten Ressourcen von Mobiltelefonen keine lokale Datenbank mit diesen Informationen gespeichert werden kann.

Idealerweise sollte der erwähnte „Olivanda-Client“ auf vielen verbreiteten Mobiltelefonen funktionieren, damit die potenzielle Benutzergruppe groß gehalten wird. Die Verwendung der Java Platform, Micro Edition („J2ME“) ermöglicht dies prinzipiell, da fast alle modernen Mobiltelefone Java-Unterstützung bieten.

Durch die hohe Heterogenität bezüglich der Ein- und Ausgabemethoden wird der Arbeitsaufwand bei Entwicklung für mehrere Telefontypen jedoch erheblich gesteigert. Deswegen sollte sich die erste Version des Olivanda-Clients auf Telefone vom Typ Nokia N95 beschränken. Zudem wurde das Tool J2ME-Polish eingesetzt, das die Entwicklung von mobilen Anwendungen deutlich vereinfacht.

3.2. Grafische Gestaltung des Handy-Clients

Bei der Gestaltung des Handy-Clients wurden folgende Überlegungen angestellt. Die GUI sollte

- ansprechend sein,
- wichtige Elemente hervorheben und untereinander abgrenzen,
- durch Icons den Wiedererkennungswert einzelner Funktionen verstärken,
- dem Corporate Design der Projektgruppe entsprechen,

- Funktionen mit wenig Klicks erreichbar halten.

Wir verwenden als Hintergrundbild das Logo der Projektgruppe. Im Vordergrund stehen die einzelnen Menüpunkte in einer Liste. Jeder Punkt hat ein vorangestelltes Icon, welches zusätzlich ein Indikator für die Funktion des Menüpunktes sein soll. Bei häufiger Benutzung kann so die Funktionalität eines Menüpunktes auf einen Blick anhand des Icons erkannt werden.

Jede „Seite“ des Clients hat einen Titel, der in einer Leiste über dem Menü steht. Hier kann der Benutzer jederzeit sehen, an welcher Stelle des Programms er sich befindet.

Unterhalb des Menüs befindet sich eine Navigationsleiste, die in der Regel die Buttons „Auswählen“ und „Zurück“ belegen. Mit dem rechten Hotkey (Zurück, Beenden) kommt der Benutzer auf die vorherige Seite zurück oder beendet das Programm wenn er sich im Hauptmenü befindet. Der linke Hotkey, wie auch der mittlere Button, dient der Auswahl von Elementen, sowie der Bestätigung. Die genaue Belegung innerhalb der verschiedenen Seiten wird immer angezeigt.

Die genaue Menüführung ist wie folgt:

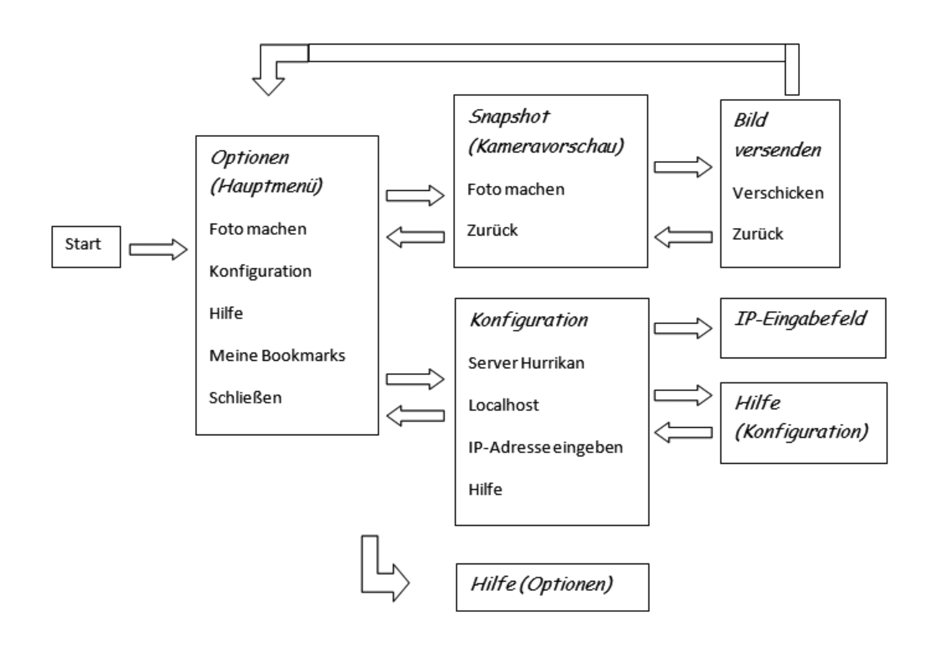


Abbildung 3.1.: Navigationsstruktur des Clients

3.3. Bildanalyse

Das Ziel der Bildanalyse im Kontext von OLIVANDA war es, die eindeutige Identifizierung von bekannten Objekten zu ermöglichen. Hierfür potenziell geeignete Verfahren

sind digitale Wasserzeichen, Objekterkennung mittels Klassifizierung und lokale Eigenschaften von Bildern. Digitale Wasserzeichen können sehr robust sein, erfordern jedoch eine Modifikation der Werbeplakate. Diese Voraussetzung kann aufgrund unserer Projektvorgaben nicht getroffen werden, sodass dieses Verfahren kategorisch ausgeschlossen werden konnten. Die Objekterkennung über Klassifikation, beispielsweise mittels Supported-Vector-Machines oder neuronaler Netze, befindet sich derzeit noch in einem sehr experimentellen und durch Forschungsarbeiten geprägten Stadium. Ein Einsatz dieser Verfahren erschien sehr risikoreich. Daher verblieb als letzte Kategorie nur das Verfahren der lokalen Eigenschaften.

Bildvergleiche mittels lokaler Eigenschaften sind sehr verbreitet und finden in zahlreichen Gebieten Anwendung. Diese sind beispielsweise Dublettenerkennung [11], Panoramastitching [13] und Rekonstruktion von 3D-Objekten [17]. Weitere Anwendungen finden sich im Motiontracking z.B. in CAD-Anwendungen [7], im maschinellen Sehen für Robotersteuerung [15] oder in der Gesichtserkennung [4]. Sie bilden also die Grundlage dafür, Merkmale auf Fotos und in Videos zu identifizieren, auf denen anschließend Objekte erkannt werden, um mit dieser Kenntnis bestimmte verknüpfte Dienste anbieten und nutzen zu können.

Die Neurowissenschaft [21] zeigt, dass Merkmale von Objekten unabhängig von deren Größe und Drehung in den neuronalen Netzen repräsentiert werden. In diesem Kontext wird der Begriff „Invarianz“ verwendet und charakterisiert die Eigenschaft einer Merkmalsgröße, die sich dadurch auszeichnet, dass sie nach einer Transformation des zugehörigen Objektes noch dieselbe Größe aufweist [14]. Üblicherweise sind diese Transformationen bei Bildern sämtliche mögliche affinen Abbildungen sowie die üblichen Punktoperationen wie Kontrast- oder Helligkeitsänderungen. In Zusammenhang mit der Suche nach den affinen Modellparametern ist es ebenfalls nützlich, wenn eine Robustheit gegenüber Objektverdeckungen besteht [5].

Eine Menge von Schlüsselpunkten beziehungsweise Merkmalen (Keypoints oder Features) kann ein Objekt in einem Bild repräsentieren, indem sie die wesentlichen Merkmale verkörpert. Besondere Merkmale sind Intensitätsunterschiede im Bild, die auf Kanten und Ecken hindeuten. Die Angabe der Keypoint-Koordinaten ist offensichtlich nicht gegen affine Transformationen invariant. Die gewünschte Stabilität wird daher i.A. durch normierte Informationen über die Umgebung des Merkmals erreicht [20].

Die Merkmalsextraktion gliedert sich in zwei logische Schritte. Der Erste ist das Auffinden von stabilen Schlüsselpunkten und der Zweite ist die Erstellung eines invarianten Deskriptors an der Stelle des Schlüsselpunktes. Exemplarisch werden die beiden Verfahren SIFT und SURF betrachtet. SIFT ist der erste besonders leistungsfähige Vertreter dieser Art [16] und SURF gilt bei vergleichbarer Leistung als besonders schnell [2].

Prinzipiell basieren die vorgestellten Verfahren zur Extraktion der Merkmalsvektoren nur auf den Helligkeitsinformationen der Bilder, weshalb diese Verfahren farbunabhängig sind.

Vorläufer von SIFT zur Extraktion lokaler Eigenschaften, wie beispielsweise solche, die

auf dem Harris-Corner-Detector [8] basieren, weisen bezüglich der Skalierungsinvarianz erhebliche Schwächen auf. Beim Harris-Corner-Detector liegt dies daran, dass nicht alle Ecken auf unterschiedlichen Skalierungsgrößen stabil sind und der Detektor dies nicht berücksichtigt [16].

SIFT ist das erste Verfahren, welches u.a. eine Skalierungsinvarianz („Scale Invariant Feature Transform“) bietet. Nachdem es 1999 vorgestellt wurde, entstanden viele Varianten dieses Algorithmus, durch die insbesondere die Robustheit und die Geschwindigkeit verbessert werden sollten. Lowe stellte vier Jahre später eine verbesserte Variante des SIFT [18] vor, welche 2004 durch PCA-SIFT [12] und 2005 durch GLOH [19] ergänzt wurde.

Die extrahierten Merkmale weisen in ihren jeweiligen Ausprägungen der Vektorelemente unterschiedliche Varianzen auf, obwohl sie dasselbe Objekt repräsentieren. Daher müssen nun im zweiten Schritt Aussagen über die Ähnlichkeit von Vektoren gewonnen werden. Dabei bildet eine vorhandene Menge von Datensätzen die Vergleichsbasis. Diese Aufgabenstellung lässt sich auf das Problem des Auffindens des nächsten Nachbarn innerhalb einer vorgegeben euklidischen Distanz eingrenzen.

3.4. Kontextuelle Bookmarks und Personalisierung

Das Konzept der Kontexuellen Bookmarks [9] bezeichnet die Verknüpfung von einem Abbild mit den Metadaten über den repräsentierten Inhalt. Das Abbild ist im Allgemeinen ein Foto, aufgenommen von mobilen Endgeräten. Dieses Gerät verfügt dabei über eine Reihe von Sensoren, wie beispielsweise Zeit, GPS oder Temperatur, die mit dem Bild verknüpft werden. Anschließend erfolgt eine Inhaltsanalyse des Bildes, z.B. über Merkmalserkennung, Texterkennung oder Gesichtserkennung, um festzustellen, was dort abgebildet wird. Dadurch ist es möglich, das konkrete physikalische Objekt digital zu repräsentieren. Der Kontextuelle Bookmark ist somit abhängig vom Nutzer, der Situation in der er gemacht wurde, sowie dem Gerät und dem Zeitpunkt, zu dem er betrachtet wird. Der Bookmark gestattet es somit, auf eine Reihe von variablen kontextabhängigen Diensten zuzugreifen.

Im Rahmen von Olivanda erschien es ferner sinnvoll, das System durch eine Nutzerpersonalisierung zu ergänzen, um die Auswahl von kontextbasierten Services zu verbessern.

Für die Personalisierung der Auswahl der Dienste muss eine Identifikation beim Server stattfinden. Dazu vergibt der Server beim ersten Verbinden eine eindeutige Nummer für den Client, die er bei jedem nachfolgenden Verbindungsvorgang angibt. Auf diese Weise kann der Server die Clients wiedererkennen und anhand ihrer Auswahlhistorie beispielsweise eine kollaborative Filterung der Dienste durchführen. Hiermit lassen sich auch einfach weitere Metadaten des Nutzers wie Geschlecht, Hobbies oder Alter verknüpfen.

Es können entweder die Sortierungsreihenfolge der Dienste beeinflusst, oder unpopuläre Dienste aussortiert werden. Im letzteren Fall muss dem Benutzer eine Möglichkeit zur

Verfügung gestellt werden, sein Profil zurückzusetzen, um ihm den Zugriff auf bestimmte Dienste nicht permanent zu verwehren. Die Personalisierung erlaubt es somit auch, Nutzer in Interessensgruppen zu clustern, um dem Nutzern von Anfang an eine bessere Auswahl anbieten zu können.

Zu den Metainformationen, die über den Nutzer erfasst werden, gehören das Alter, das Geschlecht sowie seine Interessen. Das mobile Endgerät liefert dazu das Bild des Objektes, die (letze bekannte) GPS-Position und die Zeit. Dienste werden nur mit einer Plakatkampange in Verbindung gebracht, sodass die Dienste ebenfalls äquivalente Metadaten wie Nutzer und Bild haben müssen. Durch ein Ranking der meisten Treffer zwischen den Nutzer- und Bildmetadaten sowie der Service-Metadaten entsteht der kontextuelle Bookmark, dessen Dienste in der Reihenfolge der wahrscheinlichsten Nutzung durch den Benutzer stehen.

3.5. Client-/Server-Architektur

Wie bereits angedeutet, unterteilt sich der Arbeitsablauf zur Informationsergänzung in zwei eigenständige Anwendungen, die über eine Netzwerkverbindung miteinander kommunizieren.

Der Olivanda-Client ist für die Aufnahme der Plakatfotos vor Ort zuständig. Anschließend sendet er diese an den Olivanda-Server, der mittels dem in Abschnitt 3.3 beschriebenen SIFT-Verfahren das Plakat erkennt und die dafür verfügbaren Zusatzinformationen an den Client überträgt. Dieser lässt den Benutzer auswählen und stellt die Informationen dar.

Die Kommunikation zwischen Client und Server erfolgt dabei über TCP/IP, wobei die Netzwerkverbindung zunächst nur über WLAN hergestellt wurde. Im weiteren Verlauf des Projektes wurde es möglich, Internetverbindungen über GPRS oder UMTS zu nutzen.

3.6. Evaluation der Algorithmen

Um die verschiedenen Algorithmen zur Bilderkennung zu evaluieren, wurde ein eigenständiges Programm erstellt, welches spezielle Testsets nutzt. Die Testsets bestanden dabei aus einem Referenzbild und mit Hilfe eines Bildbearbeitungsprogrammes veränderte Versionen des Referenzbildes als Vergleichsbilder. Die Veränderungen waren z.B. Weichzeichnen, Hinzufügen von Bildrauschen, Verzerren und Verändern des Kontrastes. Die Evaluation erfolgte anschließend durch das Auswerten von Recall, Precision und Fallout [22]. Dabei konnten sowohl mehrere Algorithmen untereinander, verschiedene Auflösungen der Vergleichsbilder und verschiedene Grenzwerte, ab dem ein Bild als erkannt gilt, verglichen werden.

3.7. Erkennen von Kartenausschnitten zu Navigationszwecken

3.7.1. Konzept

Die Überlegung zu diesem Konzept sah vor, dass ein Nutzer durch eine grafische Karte wie z.B. dem Heideparkplan, sowie eines Mobiltelefons mit GPS, von seinem aktuellen Standpunkt zu einer beliebigen Attraktion navigieren möchte. Dazu macht dieser mit dem Mobiltelefon ein Foto von dieser Attraktion, schickt dieses an einen Server und bekommt anschließend eine entsprechende Route zurück.
Zur Erkennung des fotographierten Kartenausschnitts sollte unsere SIFT-Implementierung dienen und die Routenberechnung von einem anderen Server erfolgen. Damit die einzelnen Attraktionen erkannt werden können, muss die ganze Karte in kleine Ausschnitte unterteilt werden, die dann als Referenzbilder verwendet werden. Die einzelnen Kartenausschnitte stellen somit alle erreichbaren Zielpunkte dar, wobei die aktuellen GPS-Koordinaten des Mobiltelefons als Startpunkt dienen.
Allerdings können einige Probleme auftreten, wie z.B. verwackelte Bilder, weil der Plan und das Mobiltelefon in der Hand gehalten werden oder weil wegen schlechten Lichtverhältnissen die Belichtungszeit zu groß war. Ein weiteres Problem ist, dass das Mobiltelefon nicht nah genug an den Plan gehalten werden kann. Daher müssen die gemachten Fotos beschnitten werden. Allerdings ist es dann schwierig, kleinere Attraktionen auf der Karte mit der Kamera zu erfassen. Desweiteren sind die Attraktionen auf so einem Plan dicht beieinander, was die Gefahr von falsch-positiv erkannten Ausschnitten erhöht.
Um nun herauszufinden, ob das Erkennen von Kartenausschnitten mit Fotos von einem Mobiltelefon möglich ist, wurde eine Evaluation mit Hilfe eines von uns entwickelten Programms durchgeführt. Diese Evaluation wird in dem Kapitel „Evaluation der Algorithmen“ 3.6 beschrieben.

3.7.2. Evaluation

Zur Evaluation wurde ein Plan in normaler Größe (ca. DIN A4) und in vierfacher Größe verwendet. Der vergrößerte Plan sollte dabei einen großen Plan an einer Stellwand darstellen. Die entsprechenden Kartenausschnitte wurden erstellt, wobei darauf geachtet wurde, sowohl große, als auch kleine Attraktionen, sowie größere Bereíche zu verwenden. Anschließend wurden bei guten Lichtverhältnissen jeweils Bilder der Attraktionen mit dem einfachen Plan liegend, mit dem Plan in der Hand und dem großen Plan liegend gemacht. Diese Bilder wurden auf 640*480 beschnitten, so dass nur die Mitte zum Bildvergleich verwendet wird. Dann wurden mit den gemachten Fotos und den Kartenausschnitten Testsets für jeden Kartenausschnit erstellt und die Evaluation durchgeführt, wobei die umliegenden Attraktionen als True-Negative-Vergleichsbilder dienten.
Das Ergebniss war, dass die Erkennung bei großen Attraktionen problemlos funktioniert, bei kleinen Attraktionen konnte es zu einer falsch-positiven Erkennung kommen, wenn Teile der umliegenden Attraktionen mit auf dem Foto waren. Das Verwackeln bei guter

Beleuchtung ist dagegen kein großes Problem und wirkt sich nur bei sehr klein dargestellten Attraktionen aus.
Um die Ergebnisse zu verbessern, müsste man die Karten vorher so anpassen, dass die Attraktionen weiter auseinander liegen und eine entsprechende Mindestgröße aufweisen, dann funktioniert das Verfahren sehr gut. Ansonsten besteht immer eine Gefahr der falsch-positiven Erkennung. Dies kann eventuell akzeptiert werden, da die falsch erkannte Attraktion in der Nähe der gesuchten liegt. Wenn die Karten nicht angepasst werden sollen, müssen die Kartenausschnitte größer gewählt werden. Allerdings werden dadurch weniger Zielpunkte geliefert, weswegen die Navigation nicht mehr ganz so genau ist.
Zusammenfassend kann festgestellt werden, dass die Navigation mit Hilfe von Bilderkennung und Kartenausschnitten möglich ist.

4. Evaluation der Interaktionsmöglichkeiten der Anwendung

In Release 2 Iteration 3 wurde eine Evaluation der Software durchgeführt. Als Zeitpunkt für die Evaluation wurde der 28. April 2008 ausgewählt. Vor der Durchführung war die Implementierung der provisorischen Bookmarks (siehe Abschnitt 4.1) durchzuführen. Als Lokalität für die Evaluation wurde das Mensafoyer ausgewählt.

Das Mensafoyer bot uns die Vorteile, dass wir dort wetterunabhängig arbeiten konnten und die Stromversorgung für den Server und den WLAN-Router vorhanden war. Als Uhrzeit wurde die Mittagszeit angedacht, um ein möglichst großes Spektrum an Testpersonen zu erreichen.

Konzept, Durchführung und Ergebnisse der Evaluation werden in diesem Kapitel vorgestellt.

4.1. Konzept

Der Leitfaden für die Evaluation beschreibt die einzelnen Schritte, die für die Evaluation durchlaufen werden mussten. Zuerst fand eine Begrüßung statt, bei der die Projektgruppe und die Software kurz vorgestellt wurde. Es sollte erwähnt werden, dass wir aus dem Department für Informatik kommen und eine Evaluation im Rahmen der Projektgruppe durchführen, in der wir eine mobile Anwendung entwickeln, um Plakate mit Informationen aus der digitalen Welt zu verknüpfen.

Wir gaben den Testpersonen eine kurze Einweisung in die Software, mit Hilfe derer sie dann in der Lage waren, das Programm zu testen. Zur Evaluierung bekam die Testperson eine kleine Aufgabe zu lösen, um anschließend Feedback in Form eines Fragebogens zu geben. Die Durchführung sollte etwa 10 Minuten dauern.

- **Einweisung in die Software**
  Um den groben Ablauf der Anwendung zu zeigen, wurde eine kleine einfache Bildbeschreibung 4.1 erstellt, anhand derer die Applikation vorgestellt wurde.

- **Allgemeine Aufgabenstellung**
  Generelle Voraussetzungen bei der Aufgabenstellung sind:

  1. Identische Aufgabe für alle Testpersonen.

   2. Software ist gestartet und im Hauptmenü.
   3. Die IP-Adresse ist voreingestellt.

- **Aufgabe vergeben**
  Es gab drei Schritte, die zur Aufgabenstellung durchlaufen wurden.
   1. Versuchen Sie ein Foto von dem Plakat zu machen, von dem sie Informationen einholen wollen.
   2. Geben Sie uns die Information, die wir von ihnen abhängig vom Plakat verlangen.
      – Aufgabe zu CeBIT Plakat: Finden Sie die Öffnungszeiten der CeBIT und teilen Sie uns die Lösung mit.
      – Aufgabe zu KinderUni: Finden Sie heraus, welche Vorlesungen im Frühlingssemester stattfinden und teilen Sie uns die Lösung mit.
      – Aufgabe zu OUT!: Finden Sie die Kartenpreise für eine Vorstellung heraus und teilen Sie uns die Lösung mit.
   3. Kehren sie zurück zum Hauptmenü.

- **Abschluss der Evaluation** Der Fragebogen wurde an die Testperson ausgeteilt. In einem Abschlussgespräch wurden noch weitere Fragen beantwortet und auf Wunsch weiteregehende Informationen gegeben. Jeder Testperson wurde noch ein Flyer mitgegeben.

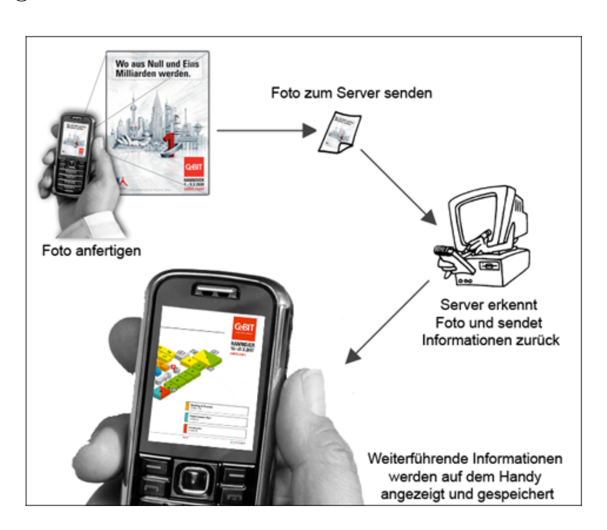


Abbildung 4.1.: Grober Ablauf der Applikation

Bookmarks zum Evaluationskonzept

Für die Evaluation wurden die folgenden drei 4.1 Plakate verwendet:







Abbildung 4.2.: Bei der Evaluation verwendete Plakate

Im Hauptmenu des Olivanda-Clients wurde ein weiterer Punkt „Meine Bookmarks“ eingefügt, unter dem eine Liste statischer Mock-Up-Bookmarks hinterlegt wurde. Vorerst waren diese Bookmarks statisch, so dass sie nicht verwaltet werden konnten. Nach dem Erkennen eines Fotos erhielt der Benutzer als Antwort die Option, den zugehörigen Bookmark einzusehen oder zurück zum Aufnahmemodus zu gelangen. Wurde das Bild nicht erkannt, erschien eine Meldung auf dem Bildschirm und man wurde zum Aufnahmemodus zurückgeleitet.

Aufbau der Bookmarks Ein Bookmark in der Bookmarkliste besteht aus einem Titel und einem Thumbnail des Originalplakats. Aus Gründen der Einfachheit beschränkten wir uns bei der Menütiefe auf zwei Schritte. Als Beispiel eines solchen Bookmarks wird hier das CeBIT-Plakat vorgestellt.

- Allgemeine Informationen
  - Was ist die CeBIT
- Anfahrtskizze
  - Straßenkarte
  - Adresse
- Messegelände
  - Hallenüberblick
- Kartenbestellung
  - Preise
  - Hotline
- Kontakt/Impressum
  - URL
  - Email
  - Telefon

- Fax
- Adresse
- Öffnungszeiten

4.2. Durchführung

Während einige Gruppenmitglieder die bestellten Aufstellwände abgeholten, aufbauten und die vorbereiteten Plakate daran befestigten, bauten die Übrigen die benötigte Hardware auf. Nach anfänglichen Problemen beim Verbinden des Clients mit dem Server auf dem Evaluationslaptop, der Bereitstellung von Strom ohne Stolperfallen für die Mensabesucher und einem Probelauf mit beiden Endgeräten, wurden die ersten Passanten angesprochen.

Während ein Gruppenmitglied vorbeilaufende Personen ansprach, führten jeweils zwei Mitwirkende für N95 und PDA die Evaluation mit einem Probanden durch. Dabei agierte ein Gruppenmitglied stets als Kontaktperson für den Probanden und erklärte das System und die Aufgaben während die zweite Person die Stoppuhr bediente und die Erfassung der Daten organisierte.

Anfänglich wurde jedem Probanden durch die Kontaktperson das Szenario erläutert, welches durch die Stellwände simuliert werden sollte und ein kurzer Einblick in das System gegeben. Daraufhin wurde die jeweilige Interaktionsart in einem Durchgang an einem vom Probanden gewählten Plakat demonstriert. Im Anschluss daran wurde der Testperson eine Aufgabe zu einem der Plakate erteilt. Dies wurde für beide Endgeräte durchgeführt und anschließend ein Fragebogen ausgeteilt.

Durch die große Resonanz während der Evaluation und das überraschend große Interesse standen durchgehend Probanden zur Verfügung und die Endgeräte waren fast durchgehend ausgelastet. So konnte die Evaluation plangemäß und reibungslos durchgeführt werden.

4.3. Ergebnisse

Die Evaluation zeigte, dass ein Großteil der Befragten den Olivanda-Client gegenüber einem herkömmlichen PDA mit Webbrowser bevorzugen würden, da sie die Bedienung als einfacher empfanden. So wurde die Bedienung des Olivanda-N95 auf einer Skala von 0 (sehr einfach) bis 4 (sehr schwierig) im Mittel mit 1 bewertet, während das PDA mit 1,5 bewertet wurde. Dies ist insbesondere auch auf die bei mobilen Endgeräten typischerweise umständliche Buchstabeneingabe – hier mittels sehr kleiner Bildschirmtastatur – zurückzuführen, wie uns von mehreren Evaluierenden bestätigt wurde.

Durch den Olivanda-Client ergab sich ein kleiner, aber statistisch signifikanter Zeitvorteil beim Erreichen der gesuchten Informationen, was neben der entfallenden manuellen

Eingabe der URL auch auf den direkteren Zugriff auf die Informationen zurückzuführen sein dürfte. Dieser wird durch die im Olivanda-Client für jeden Bookmark angebotenen Dienste ermöglicht. Während mit dem PDA im Mittel rund 65 Sekunden bis zum Finden einer Antwort vergingen, waren es mit dem N95 nur etwa 52 Sekunden.

Dennoch wäre nur ein sehr geringer Teil (in Zahlen: zwei) der Befragten bereit, für einen solchen Service zusätzlich Geld zu bezahlen. Allerdings muss hierbei auf die Zusammensetzung der Evaluierenden-Gruppe hingewiesen werden; aufgrund des Ortes war die Gruppe im Wesentlichen aus Studenten zusammengesetzt. Das mit 25 Jahren im Durchschnitt geringe Alter der Evaluierenden stützt diese Vermutung. Die finanziell meist eingeschränkten Studenten sind zu „unnötigen" Zusatzausgaben – wenn sie auch gering sein mögen – wahrscheinlich nicht bereit oder in der Lage. Dies könnte sich bei arbeitenden, finanziell besser gestellten Personen mit wenig Zeit jedoch anders darstellen. In einem solchen Szenario kommt auch der Aspekt des Fotos bzw. kontextuellen Bookmarks als Erinnerungshelfer besser zur Geltung als bei der Evaluation.

Desweiteren gaben knapp 35% der Befragten an, schon einmal zusätzliche Software auf ihrem Mobiltelefon installiert zu haben. Sie wären potentielle Nutzer des Olivanda-Projekts, für das eine Installation der Clientsoftware auf dem eigenen Mobiltelefon unumgänglich ist. Auffällig war bei dieser Frage eine starke Gliederung nach Geschlechtern. Es waren fast ausschließlich Männer, die die Frage mit „Ja" beantworteten.

5. Evaluation der Dienste

Nachdem in der ersten Evaluation der Software festgestellt wurde, dass das „Point and Shoot“-Verfahren der manuellen Eingabe gegenüber überlegen ist, blieb die Frage offen wie es um die generelle Akzeptanz des Programms stand. Die benötigten Antworten sollte eine zweite Evaluation bringen, die ihren Schwerpunkt auf die angebotenen Dienste, und weniger auf die Interaktionsart legt. Als weiterer Gesichtspunkt sollte die Geschwindigkeit der Datenübertragung überprüft werden. Zeitpunkt dieser neuen Evaluation war der 09.07.2008. Nachdem ursprünglich zwei Termine und Plakate evaluiert werden sollten, erwies sich der zweite Standort als zu gering frequentiert um dort zu evaluieren, so dass sich die Projektgruppe auf eine Evaluation mit dem Plakat der Stadt Oldenburg beschränkt hatte. Aus wettertechnischen Gründen war der erste Evaluationstag am 08.07.2008 ebenfalls weggefallen.

5.1. Konzept

Als Evaluationsverfahren wurde eine Programmpräsentation mit anschließendem Fragebogengestützten Interview gewählt. Aufgrund des Evaluationsstandortes auf dem Parkplatz des Reals wurden die Fragen den Partizipanten vorgelesen und die Antworten sowohl schriftlich als auch (bei Einverständnis) auf Video aufgezeichnet und später analysiert. Im Fokus der Befragung stand die Erwartungshaltung der Zielgruppe gegenüber der Software und eine Einschätzung der Nützlichkeit einer solchen Dienstleistung. Zusätzlich wurden demographische Grundlagen erhoben und den Teilnehmern die Möglichkeit gegeben Verbesserungsvorschläge und Kritik zu äußern.

5.2. Durchführung

Um die Teilnehmer auf den Themenbereich einzustimmen wurden einleitend Fragen zur Einprägsamkeit von Werbeplakaten gestellt. Nach einer einführenden Erläuterung des entwickelten Systems, wurden die Teilnehmer aufgefordert ein Foto des Plakates vor ihnen anzufertigen. Im Wesentlichen war die Aufgabenstellung und Durchführung dieser Evaluation ähnlich der ersten Umfrage (vgl. 4), jedoch stand den Teilnehmern lediglich ein einzelnes Plakat zur Verfügung und die Verbindung zum Server wurde über GPRS realisiert. Nachdem der Teilnehmer ein Foto des Plakates angefertigt und die Software dieses erkannt hatte, wurden weitere Fragen gestellt. Anschließend wurde aus rechtlichen

Gründen eine Erlaubnis zur Verwendung des Filmmaterials erfragt. Für die Auswertung wurden die Antworten sowohl schriftlich als auch auf Video festgehalten.

Bei der Durchführung wurde über den Ablaufzeitraum eine geringere Ausbeute an Testpersonen erzielt als erwartet wurde. Dies lag an diversen, vorher nicht beachteten Gründen. Vordergründig war der Standort ungünstig, da zwar stark frequentiert, jedoch viele Testpersonen in Eile aufgrund von Einkäufen waren und daher nicht ausreichend Zeit für die Evaluation mitbrachten. Weiterhin war das Wetter eher ungemütlich, was viele Leute nicht zum stehenbleiben animiert hat. Auf Nachfrage stellt sich heraus, dass durch die Nähe zum Discounter der Eindruck einer gewerblichen motivierten Marktforschungstudie entstand, was naturgemäß zu geringer Akzeptanz führte.

5.3. Ergebnisse

An der Befragung haben insgesamt elf Teilnehmer teilgenommen. Die meisten Teilnehmer waren größtenteils resistent gegenüber Plakatwerbung, beziehungsweise nahmen diese nicht bewusst wahr. Das Plakat der Stadt schnitt eher unterdurchschnittlich ab, da es uninteressant wirkte und die Botschaft nicht herüberkam.

Da bei dieser Evaluation keine Skalen benutzt sondern frei geantwortet wurde, ließen sich eher subjektive Ergebnisse ableiten. Generell stieß die Software auf Akzeptanz, als Anwendungsgebiete wurden beispielsweise Touristikinformationen, Gesellschaftliche Ankündigungen sowie Konzert- beziehungsweise Musicalplakate oder die Nutzung als Erinnerungshilfe genannt.

Auch zu der Ergononomie der Software konnten Aussagen getroffen werden. Da die Übertragung am Evaluationstag über GPRS realisiert wurde, wurde oft Feedback in Bezug auf die Geschwindigkeit der Datenübertragung gegeben. Diese wurde als störend langsam empfunden. Desweiteren wurde gewünscht die Menuführung noch etwas intuitiver zu gestalten, sowie einige „Zurück“ Knöpfe zu integrieren. Eine Testperson gab die Anregung eventuell hilfreiche Erläuterungen zur Funktion der Software über Piktogramme beziehungsweise erklärende Bilder geschickt einzubauen.

6. Zusammenfassung

In diesem Kapitel erfolgt eine zusammenfassende Darstellung über den Gesamtverlauf des Projektes.

Im ersten Abschnitt wird ein Fazit über die Erwartungen und Erfahrungen mit eXtreme Programming vorgestellt. Im darauffolgenden Abschnitt wird ein Gesamtfazit der Projektgruppe gezogen und im abschließenden Abschnitt ein Ausblick auf eine mögliche zukünftige Weiterentwicklung der in der Projektgruppe erzielten Ergebnisse gegeben.

6.1. Fazit von Erwartungen und Erfahrungen mit „eXtreme Programming"

Zu Beginn der Projektgruppe herrschte bei den Teilnehmern Neugierde, aber auch Skepsis beim geplanten Einsatz des „eXtreme Programming" als Vorgehensmodell für die Entwicklung von Olivanda.

Die Befürchtungen bei vielen waren, dass vor allem in der Vorlesungszeit das elementare Konzept des Pair-Programming und das tägliche Standup-Meeting nicht umgesetzt werden könnten. Diese Befürchtung lag darin begründet, dass XP als Vorgehensmodell für „Vollzeitentwickler" ausgelegt sei, während aufgrund parallel verlaufender Vorlesungen die Projektgruppenteilnehmer eher als „Teilzeitentwickler" anzusehen wären, die sich nur eher unregelmäßig treffen könnten. Außerdem wurde das Simplicity-Prinzip des XP kritisch betrachtet, da dadurch keine Gesamtstruktur des Systems zu Beginn der Entwicklung vorläge und somit möglicherweise viele Teile der Architektur oftmals im Laufe des Projektes angepasst werden müssten. Auch das Formulieren von Storycards wurde zunächst kritisch gesehen, da die Befüchtung bestand, dass aufgrund der nicht täglichen Treffen aller PG-Mitglieder die Weitergabe der Storycards problematisch werden könnte.

Bereits nach kurzer Zeit zeigte sich, dass die Skepsis bezüglich der Umsetzung von XP an einigen Stellen berechtigt war. Regelmäßige Standup-Meetings konnten erst in der vorlesungsfreien Zeit durchgeführt werden, da nur dort alle Mitglieder regelmäßig zu bestimmten Zeiten erscheinen konnten. Nach den ersten Iterationen zeigte sich außerdem, dass regelmäßig umfangreiche Refactoring-Maßnahmen sowohl im Server, als auch im Client nötig wurden. Diese waren erforderlich, da sich zeigte, dass die Umsetzung von neuen, in den Planning Games beschlossenen Features sich nicht in der zuvor konzipierten Struktur umsetzen ließen. Als Beispiel sei hier genannt, dass im Zuge der Einführung der Personalisierung jedem Olivanda-Handy-Client eine eindeutige User-ID zugewiesen

werden musste, die dieser bei einer Anfrage an den Server stets mitschicken musste. Dadurch musste das gesamte Konzept der Kommunikation zwischen dem Handy und dem Server neu konzipiert werden, da in den ersten Iterationen aufgrund des „Simplicity“-Konzeptes des XP lediglich die Möglichkeit umgesetzt wurde, ausschließlich ein Bild an den Server zu senden. Da aber bereits zu Beginn der Projektgruppe offensichtlich war, dass eine eindeutige Identifikation der Handy-Clients im Zuge der Personalisierung nötig werden würde, hätte diese bereits von Anfang an berücksichtigt werden können, auch wenn für den für die ersten Iterationen geplanten Feature-Umfang diese Identifikation nicht relevant gewesen wäre. Das „Simplicity“-Prinzip verhinderte daher die frühe Einführung der User-ID auf Client-Seite und führte zu den genannten umfangreichen Refactoring-Maßnahmen.

Ein weiteres Problem zeigte sich bei der Umsetzung des zu Beginn des Projektes aufgestellten globalen Zeitplanes. Dieser musste im Laufe des Projektes mehrmals angepasst werden, da aufgrund der zu Beginn nicht einberechneten Evaluationen des Systems zunächst anvisierte Meilensteine in Form von Iterations- und Releaseabschlüssen nicht eingehalten werden konnten.

Weiterhin fiel es der Gruppe schwer, Aufwandseinschätzungen für bestimmte Aufgaben zu definieren, wodurch schwer eingeschätzt werden konnte, welche genauen Funktionalitäten innerhalb einer Iteration bzw. eines Releases zeitlich umgesetzt werden könnten. Dies erschwerte die Aufstellung eines konkreten Planes bzgl. des Abschlusses einzelner Storycards, besserte sich allerdings mit Einführung der „Delphi-Methode“ zur Aufwandseinschätzung für die Tasks auf den Storycards. Ein Soll-Ist-Vergleich zwischen geschätzten und tatsächlich benötigten Stunden für die einzelnen Tasks erleichterte im späteren Verlauf des Projektes eine immer besser werdende Prognose für den benötigten Aufwand für konkrete Tasks, so dass zum Projektende alle geplanten Storycards vollständig innerhalb der anvisierten Zeitspanne erfüllt werden konnten.

Die Skepsis bezüglich der schlechten Umsetzbarkeit des Pair-Programmings hatte sich allerdings nicht bestätigt. Trotz der Befürchtung, dass die Implementierung aufgrund der unflexibleren individuellen Programmierzeiten langsamer voranschreiten würde, waren alle Teilnehmer der Projektgruppe zum Abschluss des Projektes von den Vorteilen des Pair-Programming überzeugt. Vor allem die bessere Code-Qualität und die Tatsache, dass durch wechselende Teams jeder einen Einblick in nahezu alle Teile des Projektes erlangte, überzeugte die Teilnehmer.

Um das Problem der Weitergabe und der fortlaufenden Ergänzung der Tasks durch der der jeweiligen Storycard zugeteilten Gruppe zu lösen, wurde vom Projektmanagement zunächst ein webbasiertes, für eXtreme Programing konzipiertes Projektmanagementtool eingesetzt. Es zeigte sich jedoch schnell, dass dieses u.a. bedingt durch die schlechte Bedienbarkeit nicht unsere Anforderungen erfüllte. Daraufhin wurden die Storycards auf Papier-Karteikarten verfasst, wobei das Zweierteam, das die jeweilige Storycard bearbeitete, diese bekam und um weitere, bei der Entwicklung hinzugekommene Tasks ergänzte. Da allerdings im weiteren Verlauf gefordert wurde, dass alle Storycards auch im Abschlussbericht aufgeführt werden sollten, wurden diese fortan nicht mehr in Pa-

pierform, sondern gleich im Abschlussbericht verfasst und ergänzt. So löste sich auch das Problem mit der Weitergabe der Storycards.

Die durch das XP vorgegebene iterative Entwicklung zeigte sich besonders vorteilhaft beim Schreiben des Abschlussberichtes. So blieb es den Teilnehmern erspart, in den letzten Wochen der Projektgruppe als letzte verbleibende und wenig abwechslungsreiche Aufgabe den Abschlussbericht vollständig zu erstellen. Während der Entwicklungszeit wurde es von einigen Teilnehmern sogar als positive Abwechslung angesehen, anstatt von Implementierungsaufgaben für einige Zeit am Abschlussbericht zu arbeiten. Zum Abschluss fielen daher nur noch Restarbeiten in Form von Rechtschreibkorrekturen und dem Erstellen einzelner neuer Abschnitte an.

Insgesamt war die Nutzung des XP als Vorgehensmodell eine lehrreiche Erfahrung, zeigte aber auch dessen Schwächen. Es ist allerdings sehr wahrscheinlich, dass die Umsetzung von XP in einer Gruppe, in der bereits einige Teilnehmer Erfahrung im Umgang mit XP haben, mit weitaus weniger Startschwierigkeiten zu rechnen gewesen wäre. Dies betrifft vor allem die Schwierigkeiten mit dem „Simplicity“-Prinzip. Möglicherweise hätten mit XP erfahrene Teilnehmer eine Lösung gesehen, wie sich dieses „Simplicity“-Prinzip mit dem Wunsch, bereits absehbare Architekturdesign-Entscheidungen, die erst in späteren Iterationen relevant werden würden, vereinbaren ließe.

6.2. Gesamtfazit der Projektgruppe

Nachdem ein Großteil der Gruppe bereits im Software-Projekt erfolgreich miteinander gearbeitet hatte, ist auch die jetzige Zusammenarbeit überwiegend reibungslos von statten gegangen. Das Arbeitsklima innerhalb der Gruppe war stets freundschaftlich und ausgeglichen. Das gemeinsame Arbeiten in den Räumen der Universität während der vorlesungsfreien Zeit verlief erfolgreich und war wesentlich produktiver als die Heimarbeit während der Vorlesungszeit.

Aufgrund des verwendeten, flexiblen Vorgehensmodells war die Projektgruppe in der Lage, sich spontanen Anforderungsänderungen und terminlichen Verschiebungen, wie sie beispielsweise durch die Evaluationen entstanden, flexibel anzupassen und diese Herausforderungen ohne größere Probleme zu bewältigen.

Obwohl gerade in der zweiten Evaluation einige Probanden die Sinnhaftigkeit des Systems durchaus in Frage stellten, vermittelte es ein gutes Gefühl innerhalb der Gruppe ein stabiles und robustes System dieser Größenordnung vom ersten konzeptuellen Gedanken bis zur fertigen Distribution selber konzipiert, implementiert und evaluiert zu haben. Gerade auch die Auswahl an Einsatzgebieten von Bildanalyse-, Webtechnologien und eigenen Konzepte ergab eine erfreuliche Vielfalt und Abwechselung während der Arbeit an Olivanda sowie einen großen Lerneffekt.

Auf Basis der Aussagen der Evaluationsteilnehmer bleibt festzuhalten, dass ein kommerzieller Einsatz der Software wegen mangelnder Abonenntenzahlen vermutlich nicht profitabel wäre.

6.3. Ausblick

Die in der Projektphase enstandenen Prototypen der Client- sowie der Serveranwendung bieten vielfältige Weiterentwicklungsmöglichkeiten. Zunächst ist eine Integrierung von Textdetektion bzw. Texterkennung sowie ein Musikinformations-Dienst denkbar. Desweiteren könnten bei einer Weiterentwicklung hin zu einem produktiv einsetzbaren System Standarddienste für Anbieter definiert werden (wie z.B. „Ticketkauf“ oder „Ticketreservierung“). Auf technischer Seite würde sich ein Umstieg der Client-/Serverkommunikation hin zu XML-Remote-Procedure-Call (xml-rpc) anbieten, da zum Ende der Entwicklung des Systems der Client dahingehend erweitert wurde, dass er Funktionalitäten des Servers über XML-Kommunikation ansteuert. Dieses könnte durch das in J2MEPolish bereits integrierte xml-rpc Framework ausgetauscht werden. Um eine standardisierte Schnittstelle zur Serveranwendung zu bieten und die Wiederverwendbarkeit zu erhöhen, wäre eine Realisierung des Servers als Webservice in Betracht zu ziehen. Abschließend bleibt noch anzumerken, dass die entstandenen Prototypen für eine erneute Evaluation genutzt werden könnten. Mit publikumswirksameren Plakaten (z.B. von bekannten Kinofilmen) wäre es evtl. möglich, aussagekräftigere Ergebnisse als in der zweiten, innerhalb der Projektphase durchgeführten, Evaluation zu erhalten.

Teil II.

eXtreme Programming

7. Einleitung

Dieser Teil des Endberichts dokumentiert die Anwendung und den Verlauf des Extreme Programming-Vorgehensmodells bei der Softwareentwicklung innerhalb der Projektgruppe OLIVANDA. Die Struktur des Berichts orientiert sich dabei am Vorgehensmodell selbst – analog zu den im Entwicklungsprozess stehenden Releases und darin enthaltenen Iterationen ist auch die Dokumentation aufgebaut. Dies soll zum einen die Nachvollziehbarkeit des Entwicklungsprozesses ermöglichen und ist andererseits der iterativen Entstehungsgeschichte des Berichts geschuldet.

Die Dokumentation eines Releases umfasst dabei jeweils einen allgemeinen Teil, der die Ziele und die Planung für das entsprechende Release beschreibt, sowie die Dokumentation der bis zu drei Iterationen, die innerhalb eines Releases stattfanden. Eine Iteration dauerte dabei in der Regel vier Wochen, allerdings waren teilweise Anpassungen an die zeitlichen Rahmenbedingungen nötig.

Im Abschnitt einer jeden Iteration sind auch die zugehörigen Storycards beschrieben, d.h. jene Storycards, für die zum Zeitpunkt der Iteration offene Tasks bestanden. Ausnahmen sind die Storycards bezüglich der Pflege des Endberichts und der Website, die dauerhaft zu bearbeiten waren.

8. Release I (Zeitraum: 07.01.2008 bis 30.01.2008)

8.1. Leitszenario

Im Rahmen des ersten Releases wurden zunächst die elementaren Strukturen geschaffen, um mit einem mobilen Endgerät ein Plakat zu fotografieren, diese Aufnahme an den Server zu senden und eine statische Information zu dem dazugehörig erkannten Plakat auf dem mobilen Endgerät angezeigt zu bekommen. Parallel hierzu fand eine Einarbeitung und Eingewöhnung der Teammitglieder in das Vorgehensmodell XP statt.
Hierdurch wurde ein Grundstein gelegt, um sich im weiteren Verlauf auf die langfristigen Ziele der Projektgruppe wie die Konzepterstellung und Implementierung von kontextuellen Bookmarks zu konzentrieren. Der Stand innerhalb des globalen Projektzeitplanes (vgl. 1.6) zu Beginn von Release I wird in Abb. 8.1 durch blaue und lila hinterlegte Zeilen verdeutlicht.

Kalenderwoche	Release	Iteration	Woche
(01.01.2008)	1	1	1
3	1	2	1
4	1	3	1
5	1	4	1
(04.02.2008)	2	1	1
7	2	1	2
8	2	1	3
9	2	2	1
10	2	2	2
11	2	2	3
12	2	2	4
13	2	3	1
14	2	3	2
15	2	3	3
16	2	3	4
(21.04.2008)	3	1	1
18	3	1	2
19	3	1	3
20	3	1	4
21	3	2	1
22	3	2	2
23	3	2	3
24	3	2	4
25	3	3	1
26	3	3	2
27	3	3	3
28	3	3	4
(14.07.2008)	4	1	1
30	4	1	2
31	4	1	3
32	4	1	4
33	4	2	1
34	4	2	2
35	4	2	3
36	4	2	4
37	4	3	1
38	4	3	2
(30.09.2008)	4	3	3

Abbildung 8.1.: Aktuelle Zeitplanung Release I

8.2. Zeitplan

Der Zeitplan für dieses Release wird in Abb. 8.2 grafisch dargestellt. Release I bestand aus vier Iterationen, wobei jede Iteration eine Woche dauerte. In jeder Woche gab es Aufgaben aus dem Bereich der Planung und Dokumentation, sowie Entwicklung der Anwendung (in Abb. 8.2 grün und blaugrau markiert). Die Aufgaben der Projektbetreuer sind in einer eigenen Zeile extra aufgeführt, während die anderen Aufgaben allgemein in den Aufgabenbereich der Projektgruppenteilnehmer fielen.

Release n	Iteration 1		Iteration 2	Iteration 3	Iteration 4	
Betreuer	Planning-Game					Akzeptanz-Tests
Planung&Doku		Doku			Doku / Präsentation	
Entwicklung		Implementierung				

Abbildung 8.2.: Zeitplanung für Release I

- Iteration 1: Übertragung Server und Handy
  Start: 07.01.2008

- Iteration 2: Plakat übertragen, auswerten und Informationen zurücksenden
  Start: 17.01.2008

- Iteration 3: Ergonomische Verbesserungen
  Start: 26.01.2008

8.3. Storycards

Im nachfolgenden Abschnitt werden die im gesamten Release I verwendeten Storycards aufgelistet, um im Gesamtkontext des Endberichtes einen Überblick über den Projektfortschritt zu geben.

In diesem Release wurden zunächst die Storycards „Projekt-Dokumentation“, „Aufnahme eines Fotos mit dem Handy“, „Grundlage der Informationspräsentation“ und „Technologie-Evaluation“ verwendet.

Projekt-Dokumentation

Lfd-Nummer: 1		
Projekt-Dokumentation		
Beginn: 01.10.2007	Risiko: niedrig	Zeit: bis Projektende
Beschreibung		
Die verschiedenen Aspekte der Projektarbeit können in einer Dokumentation nachgelesen werden. Diese kann online eingesehen werden. Die Projekt-Homepage bietet einen aktuellen und optisch ansprechenden Überblick über den Status der Projektarbeit. Der Projekt-Endbericht wird fortlaufend iterativ erweitert.		
Tasks	Verantwortlich	Erledigt
Klassendiagramm als rückblickende Übersicht	Jan	X
Grundgerüst des Endberichts	Christian M.	X
Benutzerhandbuch	nicht zugeordnet	
Online-Hilfe	nicht zugeordnet	

Aufnahme eines Fotos mit dem Handy

Lfd-Nummer: 2		
Aufnahme eines Fotos mit dem Handy		
Beginn: 23.01.2008	Risiko: hoch	Zeit: 65 h
Beschreibung		
Dem Benutzer wird innerhalb der Anwendung mit einer intuitiven Menüführung die Möglichkeit aufgezeigt, die Handykamera zu bedienen. Sucherfunktion und Auslöser gliedern sich optisch ansprechend in die Benutzeroberfläche ein und erlauben so das Aufnehmen des Kamerabildes. Im Anschluss an die Aufnahme beginnt ein Dialog zur Bildverwaltung und/oder Informationsabfrage.		
Tasks	Verantwortlich	Erledigt
Handy-User-Abfrage	Jeschua, Dennis, Torben	X
Handy-API (speichern, fotografieren…	Jan, Christian N.	X

Grundlage der Informationspräsentation

Lfd-Nummer: 5		
Grundlage der Informationspräsentation		
Beginn: 07.01.2008	Risiko: normal	Zeit: 65 h
BESCHREIBUNG		
Der Nutzer kann die gelieferten Informationen auf einem Handydisplay ansehen. Die Darstellung wird auf übliche Auflösungen optimiert und optisch ansprechend, sowie übersichtlich dargeboten.		
TASKS	VERANTWORTLICH	ERLEDIGT
Icons auswählen	Jeschua, Oliver	X
Text neben Icons setzen	Jeschua, Oliver	X
Tests	Jeschua, Oliver	X
Kamera-Display anpassen (zurück statt schließen)	Jeschua, Oliver	X
Stil des Titels anpassen	Jeschua, Oliver	X
Stil der Bedienleiste anpassen	Jeschua, Oliver	X
Hilfemenü soll von unten hochsliden	Jeschua, Oliver	
Fehlerbehebung	Jeschua, Oliver	

Technologie-Evaluation

Lfd-Nummer: 7		
Technologie-Evaluation		
Beginn: 07.01.2008	Risiko: normal	Zeit: 65 h
BESCHREIBUNG		
Die Anwendung soll möglichst die neusten Konzepte und State-of-the-art-Lösungen aus den Bereichen Personalisierung, Kommunikation etc. verwenden. Dazu werden diese auf eine mögliche Verwendung im Rahmen des Projekts geprüft.		
TASKS	VERANTWORTLICH	ERLEDIGT
Sift: Verfahrenswahl, Einarbeitung	Torben	X
J2MEPolish: lauffähig installieren	Jeschua	X
J2MEPolish: bestehendes Projekt portieren	Jeschua, Oliver, Marcel, Jan	X

8.4. Umsetzung

8.4.1. Betrachtete Algorithmen

Aufgrund der höheren Verbreitung und Evaluierungsergebnissen durch Dritte fiel die Wahl des Bilderkennungsverfahrens auf die aktuelle SIFT-Variante. Der Ähnlichkeitsver-

gleich sollte vorerst auf einem Brute-Force-Verfahren basieren. Dies war für den ersten Release angemessen, da hier keine große Referenzbasis von Bildern gefordert war.

Die Projektgruppe entschied sich dazu, die Programmiersprache Java zu verwenden. Sie wollte sowohl server- als auch clientseitig plattformunabhängig sein, was mit Java möglich war. Daraus folgte, dass auch die Implementierungen der Algorithmen in Java sein sollten. Die Anwendung inklusive der Algorithmen sollte u.a. auf einem Solaris-RISC Server lauffähig sein. Da dort keine Administrationsmöglichkeiten gegeben waren, schränkte dies die Installation von notwendigen Entwicklungswerkzeugen, die zur Kompilierung von vorhandenen nativen SIFT-Implementierungen nötig wären, erheblich ein. Kompilierungen unter Windows, häufigste Entwicklerplattform in der Projektgruppe, schlugen für SiftPP[1] und SiftGPU[2] fehl. Erfolgreich unter Windows kompilierbar, jedoch nicht auf dem gegebenen Server einsetzbar waren die C# libSift(Autopano)[3]-Bibliothek sowie die OpenCV-Sift[4]-Implementierung.

8.4.2. Evaluationen und deren Ergebnisse

Der Fortschritt der Software war zu diesem Releasezeitpunkt noch nicht umfangreich genug, daher wird in diesem Release auf eine Evaluation verzichtet. Durch die testgetriebene Entwicklung mit JUnit4 serverseitig beziehungsweise JMUnit clientseitig wurde die Funktionalität der Implementierung weitgehend gewährleistet.

8.4.3. Besonderheiten bei der Implementierung

Eine Java-Implementierung für SIFT war nicht auffindbar, sodass für SiftPP und libSift Java-Portierungen begonnen wurden. Diese erfüllten jedoch nicht die Tests. Als Alternative wurde daher ein Windows-Kompilat für OpenCV als SIFT-Merkmals-Extraktor, sowie zur Bestimmung ähnlicher Bildmerkmale verwendet. Diese ausführbare Datei war über Wine ebenfalls auf x86er Unix-Systemen lauffähig, jedoch nicht auf dem RISC-Server. Die ausführbare Datei konnte nur zwei Bilder gleichzeitig verarbeiten, so dass dafür gesorgt werden musste, dass diese Situation nie auftrat. Das Ergebnis, die Anzahl der übereinstimmenden Treffer, wurde von Java als Exitcode ausgelesen.

8.4.4. Systemarchitektur

Das System „Olivanda“ unterteilt sich in zwei eigenständige Anwendungen, die über eine WLAN-Verbindung miteinander kommunizieren. Die Abbildungen 8.3 und 8.4 stellen den Stand der Entwicklung zum Ende des Releases 1 dar.

[1] http://vision.ucla.edu/ vedaldi/code/siftpp/siftpp.html
[2] http://cs.unc.edu/ ccwu/siftgpu/
[3] http://user.cs.tu-berlin.de/ nowozin/libsift/
[4] http://web.engr.oregonstate.edu/ hess/

Im Nachfolgenden wird ein kurzer Überblick über die beiden Anwendungen gegeben. Eine detaillierte Beschreibung der einzelnen Klassen findet sich im zum Projekt gehörenden Javadoc. Der Handy-Client des Projektes teilt sich in die Pakete „handyui"„connection" und „help". „handyui" beeinhaltet dabei alle Funktionen für die Anzeige der Anwendung, „connection" stellt alle Dienste zur Netzwerkkommunikation zur Verfügung und schließlich beinhaltet das Paket „help" alle Klassen zur Umsetzung einer Benutzerhilfe.

Die eigenständige Serveranwendung umfasst alle Klassen zum Management von Clientverbindungen sowie die Pakete „analyse" und „util". Das zentrale Paket „analyse" kapselt den Zugriff auf den verwendeten SIFT-Algorithmus zur Bildanalyse. In „util" liegen Klassenstrukturen zur temporären Datenhaltung, sowie für die Anwendung definierte Exceptions.

Sowohl für den Server, als auch für den Handy-Client bestanden derzeit für alle testbaren Klassen JUnit-Tests zur Verifikation ihrer Funktionalität. Eine Ausnahme bildete das Paket „handyui". Es enthielt keine automatisierten Testmöglichkeiten, da es sich bei den darin enthaltenen Klassen um Elemente zur grafischen Anzeige des User Interfaces handelte.

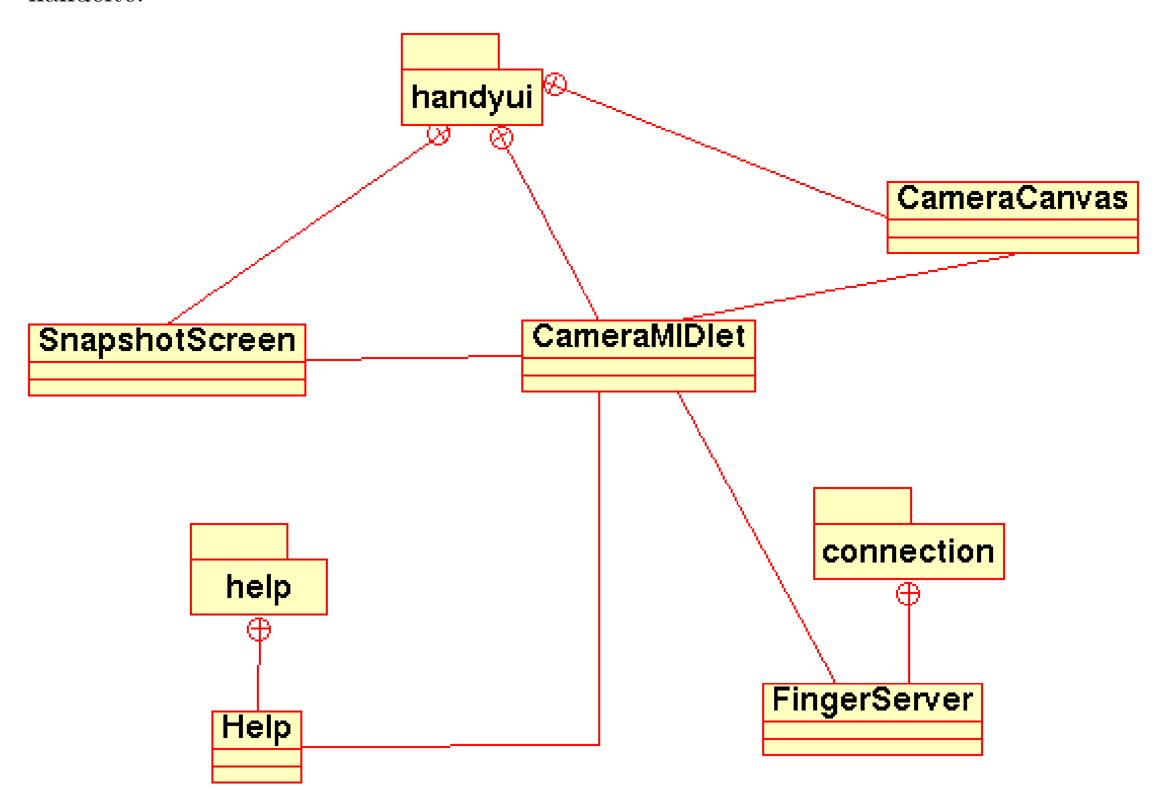


Abbildung 8.3.: Klassendiagramm des Handy-Clients

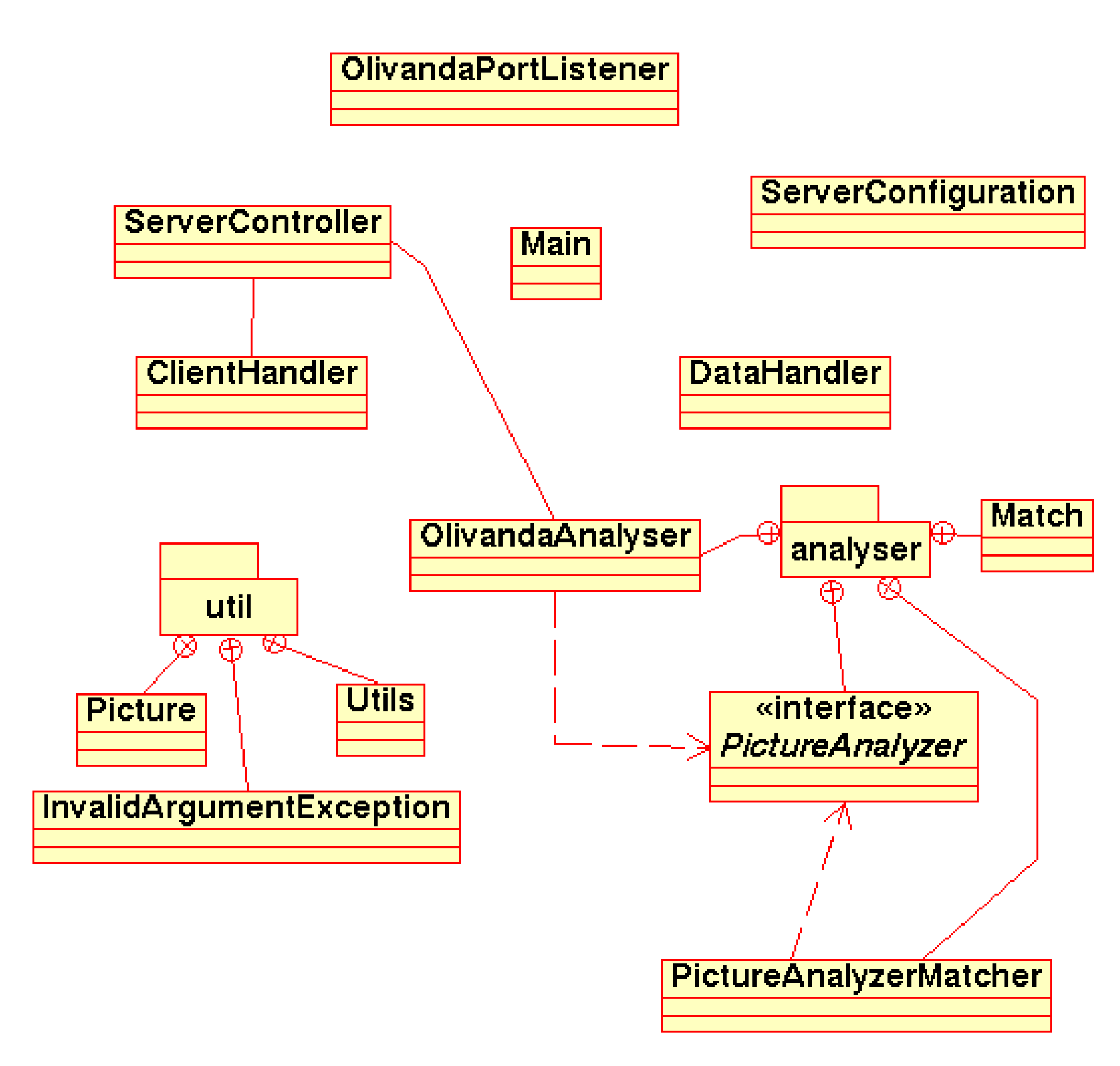


Abbildung 8.4.: Klassendiagramm des Servers

8.5. Ergebnisse

8.5.1. Retrospektive

Die Arbeit am ersten Release war in erster Linie von der Einarbeitung in das Vorgehensmodell XP geprägt. Vorgaben wie „Pair Programming“ und die testgetriebene Entwicklung waren für alle Gruppenmitglieder unbekannt und zogen so eine ganze Reihe von Erwartungen, aber auch Skepsis mit sich:

> Durch Anwendung von XP könnte mir diese Skizze [zum Erlangen der Architektur-Übersicht] im Laufe des Projektes fehlen. - Jan Steinke

> Testen mit JUnit ist für mich ganz neu und wird sicher auch seine Tücken haben. - Marcel Schröder

Die Erwartungen an XP waren innerhalb der Gruppe sehr unterschiedlich, dennoch sahen fast alle Gruppenmitglieder das Hauptproblem im Unterschied zwischen dem studentischen Arbeitsumfeld und dem Arbeitsalltag, für den XP entwickelt wurde. Diese Befürchtungen erwiesen sich als berechtigt, da es innerhalb der Vorlesungszeit eine große Herausforderung darstellte, in gemischten Vorlesungsplänen noch gemeinsame Freiräume für Pair-Programming-Sitzungen zu finden. Die von Kent Beck vorgeschlagenen „Standup-Meetings“ stellten sich in dieser Zeit ebenso als undenkbar heraus und auch das Wechseln der Team-Partner war in den meisten Fällen nicht realisierbar. Daher war die Rotation der Programmierpartner und Anwendungsgebiete, die ein wesentlicher Bestandteil der kollektiven Verantwortlichkeit ist, nicht gegeben und es entstanden wochenweise „Server-Teams“ und „Client-Teams“.

Durch die Erfahrungen mit den klassischen Vorgehensmodellen im Software-Projekt und anderen Modulen stellte auch die Einhaltung des Simplicity-Grundsatzes ein Problem dar. Stets nur das Benötigte und dies so einfach wie möglich zu entwickeln stand häufig im Konflikt mit möglichen Vorbereitungen für zukünftige Anforderungen wie beispielsweise die Internationalisierung.

Sich durch unterschiedlichen Kenntnisstand ergebende Probleme waren eine Befürchtung, die sich im Laufe des ersten Releases als größtenteils unbegründet erwies. Durch die regelmäßig, wenn auch selten wechselnden Programmiererteams entstand größtenteils ein homogenes Verständnis für die verwendeten Technologien und ihre Einbettung in dem ersten Release.

Obwohl die Zeitschätzung sich mangels Erfahrung (besonders im Zusammenhang mit XP) als schwierig herausstellte, wurde das Releaseziel zufriedenstellend erreicht und somit eine Grundlage geschaffen, auf der in den nachfolgenden Releases aufgebaut werden konnte. Zur Erleichterung der Zeitschätzung wurde die Delphimethode eingesetzt und wurde fortan als Standardvorbereitung des Planning Games eingesetzt.

Abwahl des Projektmanagementtools „eXPlainPMT“

Im ersten Release stellte sich heraus, dass das genutzte Projektmanagementtool „eX-Plain“ umständlich in der Bedienung und sehr unübersichtlich war. Diese Unübersichtlichkeit erschwerte das spontane Eintragen von Tasks zu den einzelnen Storycards. Es wurde daher beschlossen, die Storycards ab dem nachfolgenden Release auf Karteikarten zu verfassen, wobei jede Gruppe die von ihr momentan bearbeitete Storycard erhielt und weitere Tasks, die dieser Gruppe bei der Bearbeitung einfielen, schnell ergänzt werden konnten. Das Problem, dass die Storycards aufgrund der Abwahl des Projektmanagementools nicht mehr in digialer Form verfügbar waren, wurde dadurch umgangen, dass alle von den Betreuern zur Bearbeitung ausgewählten Storycards direkt in diesen Abschlussbericht übernommen wurden.

9. Release II (Zeitraum: 31.01.2008 bis 11.05.2008)

Im zweiten Release wurde ein weiterer Schritt in Richtung der releaseübergreifenden Zielsetzungen der Projektgruppe (vgl. 2.1) unternommen. Hierbei wurde über die in Release I entstandenen Elementarstrukturen hinausgegangen und Konzepte zu kontextuellen Bookmarks und einer Evaluation mit realen Benutzern erstellt.
Diese dadurch geschaffene Basis diente für eine erste Implementierung kontextueller Bookmarks mit Mockup-Daten, sowie für Ansätze einer Restrukturierung des Clients aufgrund der in der Evaluation gewonnenen Erkenntnisse.
Die in Release II gewonnenen Ergebnisse flossen als Vorbereitung für eine umfangreichere Implementierung von kontextuellen Bookmarks und Neugestaltung des Handy-Clients in das spätere Release III ein.
Der Stand innerhalb des globalen Projektzeitplanes (vgl. 1.6) zu Beginn von Release II wird in Abb. 9.1 durch blau und lila hinterlegte Zeilen dargestellt.

Kalenderwoche	Release	Iteration	Woche
(01.01.2008)	1	1	1
3	1	2	1
4	1	3	1
5	1	4	1
(04.02.2008)	2	1	1
7	2	1	2
8	2	1	3
9	2	2	1
10	2	2	2
11	2	2	3
12	2	2	4
13	2	3	1
14	2	3	2
15	2	3	3
16	2	3	4
(21.04.2008)	3	1	1
18	3	1	2
19	3	1	3
20	3	1	4
21	3	2	1
22	3	2	2
23	3	2	3
24	3	2	4
25	3	3	1
26	3	3	2
27	3	3	3
28	3	3	4
(14.07.2008)	4	1	1
30	4	1	2
31	4	1	3
32	4	1	4
33	4	2	1
34	4	2	2
35	4	2	3
36	4	2	4
37	4	3	1
38	4	3	2
(30.09.2008)	4	3	3

Abbildung 9.1.: Aktuelle Zeitplanung Release II

9.1. Zeitplan

Die Zeitplanung für dieses Release wird in Abb. 9.2 grafisch dargestellt und wurde im Vergleich zu Release I in einigen Punkten angepasst. Release II bestand aus drei Iterationen, in der jeweils, wie in Release I, Aufgaben aus dem Bereich Planung und Dokumentation, sowie Entwicklung der Anwendung bestanden (in Abb. 9.2 gelb, grün und blaugrau makiert). Die Aufgaben der Projektbetreuer sind erneut in einer eigenen Zeile extra aufgeführt, während die anderen Aufgaben allgemein in den Aufgabenbereich der Projektgruppenteilnehmer fielen.

Release n	Iteration 1	Iteration 2	Iteration 3
Betreuer	Auswahl des Leitszenarios für Release n		Präsentation und Akzeptanztests für Release n
Planung & Doku		Dokumentationsaufbereitung von Release n-1	Dokumentationsabschluss n-1
Entwicklung		s. Iterationen	

Abbildung 9.2.: Zeitplanung für Release II

Im Vergleich zum vorherigen Release wurden die Iterationen feingranularer geplant und auf vier Wochen verlängert, um eine effektivere und genauer geregelte Arbeitseinteilung zu ermöglichen. Die ersten drei Wochen dienten zur Vorbereitung und Umsetzung von Storycards. Die vierte Woche wurde zur abschließenden Dokumentation von Ergebnissen eingeführt. Der neue Iterationsplan findet sich in Abb. 9.3 wieder. Der semantische Aufbau der Abbildung entspricht dem der Zeitplanung des gesamten Releases.

Iteration n	Woche 1	Woche 2	Woche 3	Woche 4
Betreuer	Retrospektive Auswahl von Storycards / Tasks für n Diskussion der Akzeptanztests	Vorstellung und Diskussion der Akzeptanztests		Präsentation n Akzeptanztests für n Kundeninterview n+1
Planung & Doku	Dokumentations- und Akzeptanztestaufbereitung von Iteration n-1			Dokumentations- und Testabschluss n-1
Entwicklung	Umsetzung der Story Cards von Iteration n			

Abbildung 9.3.: Zeitplanung der Iterationen für Release II

- Iteration I:
  Start: 31.1.2008

- Iteration II:
  Start: 28.2.2008

- Iteration III:
  Start: 24.3.2008

9.2. Iteration I

Die erste Iteration des zweiten Releases erstreckte sich über die Kalenderwochen sechs bis acht und lag somit vollständig innerhalb der vorlesungsfreien Zeit. In Abbildung 9.1 ist die Einbettung der Iteration in den Gesamtablauf dargestellt. In dieser Phase arbeitete die Projektgruppe von Dienstags bis Donnerstags ganztägig zusammen und versuchte eine möglichst umfassende Umsetzung der XP-Richtlinien zu etablieren. Es herrschten positive Erwartungen an die XP-Adaption und die Effektivität der Zusammenarbeit. Dennoch verhinderten oft Klausur- und Urlaubstermine vollzählige Treffen.

9.2.1. Storycards

In diesem Abschnitt werden die in Iteration 1 des Releases 2 benutzten Storycards dargestellt und bieten wiederum einen Überblick des in dieser Iteration aktuellen Projektstandes. Es wurden die Storycards „Projekt-Dokumentation“, „Plakat-Erkennung“, „Grundlage der Informationspräsentation“, „Evaluation der Algorithmen“und „Evaluation mit Menschen“ verwendet.

Projekt-Dokumentation

Lfd-Nummer: 1		
Projekt-Dokumentation		
Beginn: 01.10.2007	Risiko: niedrig	Zeit: bis Projektende
BESCHREIBUNG		
Die verschiedenen Aspekte der Projektarbeit können in einer Dokumentation nachgelesen werden. Diese kann online eingesehen werden. Die Projekt-Homepage bietet einen aktuellen und optisch ansprechenden Überblick über den Status der Projektarbeit. Der Projekt-Endbericht wird fortlaufend iterativ erweitert.		
TASKS	VERANTWORTLICH	ERLEDIGT
Klassendiagramm als rückblickende Übersicht	Jan	X
Grundgerüst des Endberichts	Christian M.	X
Benutzerhandbuch	nicht zugeordnet	
Online-Hilfe	nicht zugeordnet	

Plakat-Erkennung

Lfd-Nummer: 4		
Plakat-Erkennung		
Beginn: 23.01.2008	Risiko: hoch	Zeit: 125 h
BESCHREIBUNG		
Das System kann bekannte Plakate auf einem übermittelten Handyfoto erkennen und Plakat-spezifisch antworten. Die Bilderkennung ist robust gegen Drehung, Verzerrung, Rauschen etc.		
TASKS	VERANTWORTLICH	ERLEDIGT
Anstoßen der C-Bibliothek aus Java-Projekt	Torben, Marcel	X
Erhalt der Matching-Resultate	Torben, Marcel	X
Test des Verfahrens	Torben, Marcel	X
Präsentation erster Prototyp	Marcel	X
K-Baum zur Aufwandsoptimierung	Torben, Jan	X
K-Baum Integration in bestehendes Projekt	Torben, Dennis, Jan	X
Sift-Fehlerrate senken	Torben, Dennis	X

Grundlage der Informationspräsentation

Lfd-Nummer: 5		
Grundlage der Informationspräsentation		
Beginn: 07.01.2008	Risiko: normal	Zeit: 65 h
BESCHREIBUNG		
Der Nutzer kann die gelieferten Informationen auf einem Handydisplay ansehen. Die Darstellung wird auf übliche Auflösungen optimiert und optisch ansprechend, sowie übersichtlich dargeboten.		
TASKS	VERANTWORTLICH	ERLEDIGT
Icons auswählen	Jeschua, Oliver	X
Text neben Icons setzen	Jeschua, Oliver	X
Tests	Jeschua, Oliver	X
Kamera-Display anpassen (zurück statt schließen)	Jeschua, Oliver	X
Stil des Titels anpassen	Jeschua, Oliver	X
Stil der Bedienleiste anpassen	Jeschua, Oliver	X
Hilfemenü soll von unten hochsliden	Jeschua, Oliver	-
Fehlerbehebung	Jeschua, Oliver	X

Evaluation der Algorithmen

Lfd-Nummer: 8		
Evaluation der Algorithmen		
Beginn: 19.02.2008	Risiko:	Zeit: n.a.
BESCHREIBUNG		
Die Algorithmen werden anhand von verschiedenen Gesichtspunkten, wie Geschwindigkeit, Treffergenauigkeit und Robustheit einheitlich getestet und bewertet. Dies geschieht mit Hilfe eines festen Testmusters, welches ein Bilderset und fixe Grenzwerte beinhaltet.		
TASKS	VERANTWORTLICH	ERLEDIGT
Überlegen eines Grobkonzeptes für die Evaluation	Marcel & Sören	X
Refactoring des Konzepts mit dem Betreuer	Marcel & Sören	X
Referanzplakate organisieren	Marcel & Sören	X
Referenzbilder aufbereiten	Marcel & Sören	X
Testsets zusammenstellen	Marcel & Sören	X
Algorithmentest implementieren	Marcel & Sören	
Auswertung erstellen	Marcel & Sören	

Evaluation mit Menschen

Lfd-Nummer: 9		
Evaluation mit Menschen		
Beginn: 19.02.2008	Risiko:	Zeit: n.a.
BESCHREIBUNG		
Das Projekt soll mit Menschen in der Universität oder in der Stadt evaluiert werden. Dazu sollen die Versuchspersonen anhand einer Aufgabenstellung / Bedinungsanleitung den Olivanda-Client testen und bewerten.		
TASKS	VERANTWORTLICH	ERLEDIGT
Überlegen eines Grobkonzeptes für die Evaluation	Marcel & Sören	X
Refactoring des Konzepts mit dem Betreuer	Marcel & Sören	X

9.2.2. Umsetzung

Betrachtete Algorithmen

J2ME-Polish

J2ME-Polish bietet einige Features für die Programmierung von Anwendungen für mobile Geräte:

- GUI-Design mit CSS.
- einfache Möglichkeiten zur Lokalisierung
- Möglichkeiten der Serialisierung
- Einfache Anpassung an verschiedene Geräte
- Java 5 Syntax, sowie die Unterstützung von Array-Lists und anderen Features der Java Standard-Edition, die in J2ME nicht verwendet werden können.

Die Entscheindung für J2ME-Polish fiel relativ früh in der Entwicklungsphase. Es gab zu der Zeit aber noch keine Version, die unter NetBeans lauffähig war. Erst nach dem Release der Version 2.0 final ließ es sich in das Projekt fehlerfrei einbinden.

Besonderheiten bei der Implementierung

J2ME-Polish

Nach der Einbindung von J2ME-Polish mussten einige Änderungen am Client vorgenommen werden. Durch die veränderte Struktur und die Aufteilung von Grafik- und Programmelementen konnte der bisherige Programmcode des Clients nicht komplett in ein J2ME-Polish-Projekt übernommen werden. Es mussten die Dialogelemente, sowie der Kamerasucher angepasst werden.

Ein J2ME-Polish-Projekt gliedert sich im Gegensatz zu einem J2ME-Projekt in zwei Ordner. Der eine Ordner (source) enthält die Quelltexte des Projekts, im anderen Ordner (resources) werden sämtliche GUI-Ressourcen wie Icons, Bilder und die CSS-Datei abgelegt. Diese Struktur ermöglicht auch die Verwendung unterschiedlicher Designs durch Anlegen weiterer Ressource-Ordner. So können auch gerätespezifische Profile erzeugt werden.

Um GUI-Elemente zu formatieren, werden im Quelltext sog. „Style-Direktiven" eingesetzt, die einen CSS-Stil referenzieren. Diese CSS-Stile werden von J2ME-Polish in grafische Elemente übersetzt. Der Quelltext könnte auch jederzeit mit einem normalen Java-Compiler kompiliert werden, da die CSS-Referenzen in Kommentare eingebettet sind.

Der Quelltext wird von J2ME-Polish vorkompiliert, um die Style-Direktiven zu verarbeiten.

Systemarchitektur

Auf eine grafische Darstellung der Client- und Serveranwendung in dieser Iteration wird verzichtet, da keine zusätzlichen Klassen und Methoden hinzugefügt wurden. Es wurden vor allem bereits existierende Strukturen überarbeitet und darin enthaltene Funktionalitäten weiter implementiert.

9.2.3. Ergebnisse

In dieser Iteration wurde primär an der Integration von J2ME Polish gearbeitet sowie versucht, eine lauffähige Version des Bilderkennungsverfahrens SIFT zu schaffen. Dabei war auch die Umsetzung der XP-Richtlinien, insbesondere des Pair-Programmings und des Test Driven Developments, ein weiteres Ziel.

Da in dieser Iteration keine Akzeptanztests zu den Storycards existierten und keine der Storycards vollständig abgeschlossen wurde, konnten die Ergebnisse der Iteration nicht konkret festgehalten werden.

Retrospektive

Nach den großen Problemen beim Finden von Terminen während der Vorlesungszeit versprach die erste Iteration des zweiten Releases endlich die ersehnte, konsequente Umsetzung der XP-Richtlinien. Dementsprechend positiv war die Stimmung innerhalb der Gruppe. Die Aussicht auf herausfordernde Programmieraufgaben und die Zusammenarbeit im Team beflügelte die Gruppenmotivation.

Dennoch wurde dieser Motivationsschub schnell von den bis dato nicht gelösten Problemen mit dem WTK unter FreeBSD und den allgemeinen Schwierigkeiten bei der Einrichtung von J2ME-Polish gebremst. Die Arbeit an J2ME-Polish und der Versuch, die Handy-Emulatoren auf den ARBI-Rechnern zum Laufen zu bringen, gaben einen schockierenden Eindruck davon, wieviele Mannstunden für eigentlich als simpel eingeschätzte Aufgaben und technische Probleme aufzubringen sind.

Trotz dieser Motivationsbremse wurde ein produktiver Status erreicht und das Pair-Programming funktionierte sehr gut. Dennoch ergaben sich durch die sehr großen und umfangreichen Tasks nur wenige Wechsel innerhalb eines Tages und so eine geringe Rotation der Programmierer innerhalb der Themengebiete. Die Komplexität der Bildanalyse und der Polish-Einrichtung machte eine Spezialisierung einiger Gruppenmitglieder unumgänglich.

Auch der sehr große Dokumentationsaufwand, wie er von den PG-Richtlinien gefordert wird, ist unserer Ansicht nach nicht mit den XP-Paradigmen vereinbar. Laut XP soll man den Code und die Tests für sich sprechen lassen, während die Arbeit am Endbericht mit allen Anforderungen schon dem nachträglichen Schreiben eines Pflichtenheftes gleich kommt.

Zudem stellte sich die Annahme der verfügbaren Mehrstunden innerhalb der vorlesungsfreien Zeit als falsch heraus, da sich in den ersten Wochen viele auf Grund der Klausuren oder anderer Planungen Urlaub genommen haben. So standen effektiv kaum mehr Stunden pro Woche zur Verfügung, als dies während der Vorlesungen der Fall war. Für längere Urlaubsphasen musste zudem im weiteren Verlauf Zukunft eine Urlaubsvertretung für die rollenspezifischen Aufgaben bestimmt werden.

Bei der Arbeit mit J2ME-Polish und den SIFT-Algorithmen stellten sich zudem große Schwierigkeiten mit dem Testen heraus und so wurden Tests teilweise garnicht oder erst nach dem Implementieren geschrieben.

9.3. Iteration II

Die zweite Iteration des zweiten Releases erstreckte sich über die Kalenderwochen neun bis zwölf und stellte somit den zweiten Durchlauf innerhalb der vorlesungsfreien Zeit dar. In Abbildung 9.1 ist die Einbettung der Iteration in den Gesamtablauf dargestellt. Um das Verhältnis zwischen Dokumentation und Planung auf der einen Seite und der zur Implementierung verfügbaren Zeit auf der anderen Seite zugunsten der Implementierung zu verschieben, wurde beschlossen, diese und die folgenden Iterationen über vier statt drei Wochen laufen zu lassen, wie in Abbildung 9.3 dargestellt.

9.3.1. Storycards

Im Nachfolgenden werden die in Iteration 2 des Releases 2 benutzten Storycards aufgelistet, um den Entwicklungsstand des Projektes darzustellen. In dieser Iteration wurden die Storycards „Projekt-Dokumentation“, „Evaluation der Algorithmen“, „Evaluation mit Menschen“, „Client GUI anpassen“, „Fotoaufnahme“, „Antwortzeit optimieren“, „Aufbau kontextueller Bookmarks“, „Dienste in Bookmarks“, „Website aktualisieren“ sowie „Plakat- und Plakattiteldarstellung“verwendet.

Zu jeder Storycard wurden vom Kunden (d.h. von den Betreuern) durchzuführende Akzeptanztests zur Validierung der durch die Projektgruppe erreichten Ergebnisse definiert. Hierbei wurde auf den Akzeptanztest „Projekt-Dokumentation“ verzichtet, da sich für diese Storycard keine sinnvollen Vor- und Nachbedingungen definieren ließen und die Storycard über das gesamte Projekt hin definiert war.

Es ergaben sich einige allgemeine Vorbedingungen, die in der Regel für alle Akzeptanztests erfüllt sein mussten und der Übersicht halber nur an dieser Stelle festgehalten werden.

- Der Handy-Client muss auf einem N95 installiert und gestartet sein.
- Der Server muss gestartet und mit Referenzbildern initialisiert sein.
- Die Verbindung vom Handy zum Server muss eingerichtet sein.

Projekt-Dokumentation

Lfd-Nummer: 1		
Projekt-Dokumentation		
Beginn: 01.10.2007	Risiko: niedrig	Zeit: bis Projektende

BESCHREIBUNG
Die Dokumentation des Projektes wird durch die Verwendung von XP iterativ ergänzt und nach Abschluss des Projektes müssen die verschiedenen Aspekte der Projektarbeit in einer Dokumentation nachlesbar sein. Diese wird online einsehbar sein und die Projekt-Homepage bietet einen aktuellen und optisch ansprechenden Überblick über den Status der Projektarbeit. Der Projekt-Endbericht wird fortlaufend iterativ erweitert.

TASKS	VERANTWORTLICH	ERLEDIGT
Klassendiagramm als rückblickende Übersicht	Jan	X
Grundgerüst des Endberichts	Christian M.	X
Benutzerhandbuch	nicht zugeordnet	
Online-Hilfe	nicht zugeordnet	
Umstrukturierung des Endberichts	Marcel	X
Erfassung der Daten, die auf die Website gehören	Dennis	X

Evaluation der Algorithmen

Lfd-Nummer: 8	Priorität: 8	
Evaluation der Algorithmen		
Beginn: 19.02.2008	Risiko: Gering	Zeit: 30 h

Beschreibung
Um die Serverantwortzeit einer Clientanfrage optimal zu halten (vgl. Storycard „Antwortzeit optimieren“) und trotzdem richtige Plakate zu erkennen, ist es notwendig, die im Projekt verwendeten Algorithmen zu evaluieren. Die zur Bilderkennung betrachteten Implementierungen von Algorithmen zeichnen sich oft durch eine unterschiedliche Geschwindigkeit und auch unterschiedliche Qualität der erkannten Ergebnisse aus. Daher werden die Algorithmen anhand von verschiedenen Gesichtspunkten, wie Geschwindigkeit, Treffergenauigkeit und Robustheit einheitlich getestet und bewertet. Dies geschieht mit Hilfe eines festen Testmusters, welches ein Bilderset und fixe Grenzwerte beinhaltet.

Vorbedingung:

- Keine Vorbedingungen.

Nachbedingung:

- Der Server hat eine Ausgabe der zur Identifikation eines Testbildes benötigten Zeit, dem dazu erkannten Referenzbild und die Anzahl an übereinstimmenden SIFT-Keypoints ausgegeben.

Tasks	Verantwortlich	Erledigt
Überlegen eines Grobkonzeptes für die Evaluation	Marcel & Sören	X
Refactoring des Konzepts mit dem Betreuer	Marcel & Sören	X
Referanzplakate organisieren	Marcel & Sören	X
Referenzbilder aufbereiten	Marcel & Sören	X
Testsets zusammenstellen	Marcel & Sören	X
Algorithmentest implementieren	Marcel & Sören	
Auswertung erstellen	Marcel & Sören	
Erstellung eines Konzepts zur Evaluation der verwendeten Algorithmen und Evaluation unter Betrachtung der verwendeten Algorithmen. Dazu soll sowohl die Quantität (Anzahl der Bilder, die innerhalb der definierten maximalen Serverantwortzeit von zehn Sekunden verglichen werden können), als auch die Qualität (maximal 10% Fehlerrate bei der Erkennung) evaluiert werden.	nicht zugeordnet	

Evaluation mit Menschen

Lfd-Nummer: 9	Priorität: 1	
Evaluation mit Menschen		
Beginn: 19.02.2008	Risiko: Gering	Zeit: 116 h

Beschreibung
Die Entwickler einer Anwendung haben meist nur eine eingeschränkte und starre Sicht auf das von ihnen entwickelte Produkt und die Akzeptanz des Endkunden nach der Fertigstellung fehlt. Um eine möglichst hohe Akzeptanz unserer Software durch den Benutzer zu gewährleisten, muss die bestehende Anwendung mit Hilfe von real existierenden Personen evaluiert werden. Dadurch werden Defizite in der Bedienung noch während der Entwicklungsphase aufgedeckt und weiterer Entwicklungsbedarf des Projektes ermittelt. Hierzu wird zunächst ein Konzept entwickelt, das u.a. die zu befragende Zielgruppe, konkrete Fragebögen, sowie evtl. nötige Erweiterungen des Handy-Clients beeinhaltet.

Vorbedingung:

- Keine Vorbedingungen.

Nachbedingung:

- Ein Konzept zur Durchführung einer Evaluation mit Menschen wurde entwickelt.
- Die zu befragende Zielgruppe wurde definiert.
- Es wurde ein einheitlicher Fragebogen definiert.
- Der Handy-Client ist durch provisorische Menüs zur Darstellung von kontextuellen Bookmarks erweitert worden.

Tasks	Verantwortlich	Erledigt
Überlegen eines Grobkonzeptes für die Evaluation	Marcel & Sören	X
Refactoring des Konzepts mit dem Betreuer	Marcel & Sören	X
Produktion eines Flyers, um die Ziele der PG für externe Personen darzustellen	Marcel & Sören	X

Client GUI anpassen

Lfd-Nummer: 10	Priorität: 5	
Client GUI anpassen		
Beginn: 04.03.2008	Risiko: Gering	Zeit: 72 h

Beschreibung
Die Attraktivität einer Anwendung für einen Benutzer steigt durch eine intuitive Bedienung, sowie einer optisch ansprechende Gestaltung. Durch ständige Erweiterungen und Strukturumstellungen muss der Handy-Client fortlaufend angepasst werden. Die grafische Oberfläche des Handy-Clients wird konsequent einen einheitlichen Gestaltungsstil verwenden und ansprechend gestaltet sein. Alle Elemente des UIs werden nach demselben Stil aufgebaut, gestaltet und bedienbar sein. Der optische Aufbau wird sich an anerkannte Standards des Software Engineerings halten. So können alle Funktionalitäten innerhalb von maximal drei bis vier Schritten erreicht werden.

Vorbedingung:

- Keine Vorbedingungen.

Nachbedingung:

- Der Handy-Client besitzt in allen Kontextmenüs einen einheitlichen Gestaltungsstil.
- Alle Anwendungsfunktionen sind in maximal drei bis vier Schritten erreichbar.

Tasks	Verantwortlich	Erledigt
Ladebalken umgestalten (keine Nummern mehr anzeigen, „Sanduhr“)	Christian M. & Christian N.	X
Einen Entwurf zur grafischen Gestaltung des Handy-Clients erstellen		
Die Elemente der Handy-GUI auf eine einheitliche Gestaltung überprüfen		
Die Elemente der Handy-GUI an eine einheitlich Gestaltung anpassen		
Skalierung der aufgenommenen Fotos auf die Größe des verwendeten Handy-Displays	Christian M. & Christian N.	X

Fotoaufnahme

Lfd-Nummer: 11	Priorität: 2	
Fotoaufnahme		
Beginn: 04.03.2008	Risiko: Hoch	Zeit: 51 h

BESCHREIBUNG
Der Benutzer möchte evtl. mehrere Plakate hintereinander mit der Anwendung fotografieren und nacheinander Hintergrundinformationen dazu erhalten. So muss durch die Anwendung die Möglichkeit angeboten werden, ohne Neustart eine beliebige Anzahl an Fotos hintereinander aufzunehmen und sie jeweils an das System zu senden. Während der Übertragung zum Server und dessen Antwort wird das jeweils aufgenommene Foto als Hintergrund angezeigt. Als Resultat wird der Benutzer eine Antwortanzeige erhalten.

Vorbedingung:

- Keine Vorbedingungen.

Nachbedingung:

- Es wurden mehrere Fotos aufgenommen, jeweils an das System gesendet und der Handy-Client läuft weiterhin stabil.
- Während jeder Anfrage an den Server nach einer Fotoaufnahme wird das jeweils aufgenommene Foto als Hintergrund angezeigt.
- Nach jedem Versenden wurde dem Benutzer eine Antwortanzeige präsentiert. Dies geschieht auch, wenn ein Plakat nicht erkannt wurde.

TASKS	VERANTWORTLICH	ERLEDIGT
Der Handy-Client wird darauf angepasst, ohne Neustart beliebig viele Fotos hintereinander senden zu können, ohne sich danach aufzuhängen	Christian N. & Sören & Oliver & Jan	X
Darstellung des aufgenommenen Fotos während der Übertragung als Hintergrundbild	Sören	X
Eine Möglichkeit zur Entscheidung, ob ein Foto an den Server gesendet werden soll oder nicht, wird implementiert	Christian N. & Jan & Oliver & Sören	X

Antwortzeit optimieren

Lfd-Nummer: 12	Priorität: 3	
Antwortzeit optimieren		
Beginn: 04.03.2008	Risiko: Mittel	Zeit: 5 h

BESCHREIBUNG
Die Performanz einer Software trägt maßgeblich zu deren Akzeptanz bei den Benutzern bei. Werden Fotos zur Analyse vom Handy an den Server übertragen, führt dies allerdings in Abhängigkeit von der Geschwindigkeit der Verbindung zu einer Wartezeit bei dem Benutzer und somit u. U. zu einer mangelnden Akzeptanz. Um eine möglichst kurze Wartezeit zu erreichen, soll, basierend auf einer nicht überlasteten WLAN-Verbindung, in 95% der Fälle eine Antwort des Servers innerhalb von maximal 10 Sekunden erfolgen. Um dieses Ziel zu erreichen, wird die vom Handy übertragene Datenmenge möglichst klein gehalten und eine Optimierung des Bildanalyse-Algorithmus vorgenommen.

Vorbedingung:

- Keine Vorbedingungen.

Nachbedingung:

- In 95% der Fälle wird eine Antwortzeit unter 10 Sekunden erreicht.

TASKS	VERANTWORTLICH	ERLEDIGT
Die Aufnahmegröße eines Fotos wird angepasst, um eine effiziente Übertragung zu ermöglichen	Dennis & Christian M.	X
PictureAnalyser optimieren	Dennis & Christian M.	X

Aufbau kontextueller Bookmarks

<table>
<tr><td colspan="2">Lfd-Nummer: 13</td><td colspan="2">Priorität: 6</td></tr>
<tr><td colspan="4">Aufbau kontextueller Bookmarks</td></tr>
<tr><td colspan="2">Beginn: 04.03.2008</td><td>Risiko: Hoch</td><td>Zeit: 87 h</td></tr>
<tr><td colspan="4">BESCHREIBUNG</td></tr>
<tr><td colspan="4">Um Hintergrundinformationen und weitere Dienste zu Plakaten zu erhalten, wird der Benutzer über das Versenden von gemachten Fotos kontextuelle Bookmarks erstellen und anfragen können, die ihm weiterführende und evtl. anschließend verfügbare Dienste zur Verfügung stellen. Die Problematik besteht darin, ein Konzept zur Gestaltung und Repräsentation von kontextuellen Bookmarks zu finden und ggf. zu erstellen. Hierzu wird zunächst nach bereits existierenden Lösungsansätzen recherchiert und darauf basierend ein Konzept entworfen, wie durch den Benutzer angefragte Bookmarks in sich aufgebaut sind und wie sie ihm präsentiert werden.</td></tr>
<tr><td colspan="4">Vorbedingung:
• Keine Vorbedingungen.
Nachbedingung:
• Es wurde ein Konzept zum Aufbau kontextueller Bookmarks erstellt und in einer Präsentation vorgestellt.</td></tr>
<tr><td colspan="2">TASKS</td><td>VERANTWORTLICH</td><td>ERLEDIGT</td></tr>
<tr><td colspan="2">„State of the art“- Analyse bestehender Techniken</td><td></td><td></td></tr>
<tr><td colspan="2">Konzeptentwurf zum Aufbau eines Bookmarks</td><td></td><td></td></tr>
<tr><td colspan="2">Konzeptentwurf zur Repräsentation von Bookmarks auf dem Handy-Client</td><td></td><td></td></tr>
</table>

Dienste in Bookmarks

<table>
<tr><td colspan="2">Lfd-Nummer: 14</td><td>Priorität: 7</td></tr>
<tr><td colspan="3">Dienste in Bookmarks</td></tr>
<tr><td>Beginn: 04.03.2008</td><td>Risiko: Hoch</td><td>Zeit: n.a.</td></tr>
<tr><td colspan="3">BESCHREIBUNG</td></tr>
<tr><td colspan="3">Dem Benutzer sollen neben einfachen Hintergrundinformationen zu fotografierten Plakaten auch weitere externe Dienste und Dienstleistungen angeboten werden. So müssen die vom Benutzer angefragten Bookmarks mit externen Diensten und Dienstleistungen verknüpft werden. Hierzu muss zunächst ein Lösungsansatz zur Einbettung von Diensten und Dienstleistungen entwickelt und testweise mit Mockup-Diensten implementiert werden. Der Benutzer wird so über die für ihn gespeicherten Bookmarks Zugriff auf dazu passende Dienste/Dienstleistungen erhalten.</td></tr>
<tr><td colspan="3">Vorbedingung:
• Keine Vorbedingungen.
Nachbedingung:
• Es wurde ein Konzept zur Einbettung von Diensten und Dienstleistungen in kontextuelle Bookmarks entwickelt und in einer Präsentation vorgestellt.</td></tr>
<tr><td>TASKS</td><td>VERANTWORTLICH</td><td>ERLEDIGT</td></tr>
<tr><td>Konzept zur Einbettung von Diensten/Dienstleistungen in Bookmarks erstellen</td><td></td><td></td></tr>
<tr><td>Realisierung des erstellten Konzeptes mit Mockup-Diensten</td><td></td><td></td></tr>
<tr><td>Anpassung und Erweiterung der Realisierung auf real existierende Dienste</td><td></td><td></td></tr>
</table>

Website aktualisieren

<table>
<tr><td>Lfd-Nummer: 15</td><td colspan="2">Priorität: 9</td></tr>
<tr><td colspan="3">Website aktualisieren</td></tr>
<tr><td>Beginn: 04.03.2008</td><td>Risiko: Gering</td><td>Zeit: 20 h</td></tr>
<tr><td colspan="3">BESCHREIBUNG</td></tr>
<tr><td colspan="3">Die Website eines Projektes dient als Aushängeschild und Repräsentation in der Öffentlichkeit und stellt den aktuellen Stand einer Projektarbeit dar. Durch die ständige Weiterentwicklung der Anwendung und damit verbundener Erreichung neuer Releases und Iterationen ist es notwendig, die Daten der Website aktuell zu halten. Die Website bietet den aktuellen Entwicklungsstand der Anwendung sowie eine Darstellung des Projektes (Beschreibung, Zielsetzung, Vorgehensweise und aktuellen Entwicklungsstand) selbst.</td></tr>
<tr><td colspan="3">Vorbedingung:
• Keine Vorbedingungen.
Nachbedingung:
• Die Webseite zeigt den aktuellen Stand der Entwicklung, eine Beschreibung und die Zielsetzung des Projektes.</td></tr>
</table>

TASKS	VERANTWORTLICH	ERLEDIGT
Erfassung der Daten, die auf die Website gehören	Dennis	X
Stand des Projektes auf der Homepage grafisch darstellen	Dennis	X

Plakat- und Plakattiteldarstellung

Lfd-Nummer: 16	Priorität: 4	
Plakat- und Plakattiteldarstellung		
Beginn: 04.03.2008	Risiko: Hoch	Zeit: 54 h
BESCHREIBUNG		
Kontextuelle Bookmarks werden nicht nur aus einem Textstring bestehen und der Benutzer möchte das auf seinem Foto erkannte Plakat angezeigt bekommen. Momentan gibt der Server als Antwort auf eine Anfrage lediglich den Dateinamen des erkannten Plakates als Antwort an den Handy-Client zur Anzeige zurück. Der Benutzer der Anwendung soll als Antwort auf ein an das System gesendetes Foto eine Darstellung des auf dem gemachten Foto erkannten Plakates und den dazugehörigen Plakat-Titel angezeigt bekommen.		
Vorbedingung: • Der Benutzer hat ein Foto an den Server gesendet. *Nachbedingung:* • Der Client hat den vom Server gesendeten Plakattitel und das entsprechende Thumbnail des fotografierten Plakates angezeigt.		
TASKS	VERANTWORTLICH	ERLEDIGT
Client-Server Kommunikation in Bezug auf andere Formate als Fotos (wie Bookmarks) anpassen	Oliver, Jan	X
Anstatt des Dateinamens wird das erkannte Plakat vom Server zusammen mit dessen Titel als Antwort gesendet	Oliver, Jan	X
Der Handy-Client muss die empfangene Serverantwort entsprechend der darin enthaltenen Daten anzeigen können	Marcel, Jan	X

9.3.2. Umsetzung

Betrachtete Algorithmen

Nach der konzeptuellen Einführung in die Bildanalyze in Abschnitt 3.3 werden hier nun die dafür notwendigen Algorithmen etwas detailierter vorgestellt.

Für die Plakaterkennung ist es notwendig, serverseitig Featurevektoren effizient miteinander zu vergleichen. Da die üblichen Standardmethoden wie z.B. binäre Suche oder

Hashlisten zum effizienten Vergleich nicht auf mehrdimensionalen Daten (mehr als drei Dimensionen) anwendbar sind, haben sich entsprechende heuristische Verfahren und spezielle Datenstrukturen herausgebildet, die diesen Missstand umgehen möchten. Die geläufigsten Datenstrukturen sind hierbei k-D-Bäume und Hashverfahren.

K-D Bäume sind die allgemeine Form eines binären Suchbaumes. Dabei werden die Datenelemente jeweils an der Hyperebene, analog zur Halbierung im binären Suchbaum, der entsprechenden Dimension möglichst halbiert. Dies wird für jedes Vektorelement, also jede Dimension, nacheinander durchgeführt und das Element dann an der zugehörigen Stelle im Baum eingefügt [3]. Auf dem Baum können dann diverse Heuristiken, beispielsweise Best-Bin-First, eingesetzt werden um effizient die Nachbarn bestimmen können. Ein anschauliches Beispiel in 2D ist im Internet[1] zu finden. Ein guter Vergleich von anderen Datenstrukturen, die Baumstrukturen verwenden, ist unter [1] angeführt.

Ein mögliches Verfahren zur Dimensionsreduktion ist das „Locality-Sensitive Hashing“ [6]. Es ist eine spezialisierte Hashfunktion und ein zweistufiges Verfahren zur Dimensionsreduktion. Dabei wird in hochdimensionalen Daten eine heuristische Nachbarschaftssuche durchgeführt um die Ähnlichkeit zu ermtteln. Der Grundgedanke ist dabei, dass ähnliche Vektoren Kollisionen erzeugen. Somit müssen Vektoren, die in der Hashliste nebeneinander liegen, Nachbarn im Sinne der Ähnlichkeit sein.

Da keine stabile öffentliche Implementierung für LSH gefunden wurde, entfällt die Nutzung entsprechend, da eine Implementierung im Rahmen der PG nicht durchführbar erschien.

Im Rahmen der Projektgruppe wurde aus zeitlichen Gründen bei der Umsetzung auf die Verwendung von Best-Bin-First verzichtet. Stattdessen wird zum Bestimmen des nächsten Nachbarn die Stelle im Baum aufgesucht, an der der Vergleichsknoten eingefügt werden würde. Anschließend wird ein Hyperrechteck über diesen Bereich aufgezogen, sodass die potentiell nächsten Nachbarn sich darin befinden müssen. Die Menge der Knoten innerhalb dieses Hyperrechteckes wird anschließend linear durchsucht um denjenigen Nachbarn mit dem kürzesten Abstand als nächsten Nachbarn zu verwenden.

Evaluationen und deren Ergebnisse

Um die Anwendung unter den Gesichtspunkten Usability, Ergonomie, Qualität und Performanz zu untersuchen und verbessern, wurde eine Evaluation für sinnvoll erachtet. Die Projektgruppe kam zu dem Schluss, dass eine Trennung zwischen der Evaluation der Algorithmen, zum Analysieren der Qualität und Performanz der Anwendung und einer Evaluation an Menschen, zur Verifizierung der Usability und Ergonomie der Software nötig sei.

[1] Universität Osnabrück, URL: `http://www-lehre.informatik.uni-osnabrueck.de/~dbs/2001/skript/node34.html`, (letzter Zugriff 03.12.07)

Evaluation der Algorithmen

Für die Evaluation der Algorithmen mussten feste Rahmenbedingungen und eine Testumgebung geschaffen werden, um aussagekräftige Ergebnisse zu erhalten. Hierzu wurden Testsets von Bildern (vgl 9.4) erstellt, welche dem jeweiligen Algorithmus erschwerte Aufnahmebedingungen simulieren und so seine Robustheit überprüfen. Die Testsets beinhalten hierbei Bilder, welche wie folgt manipuliert wurden:

1. Bewegungsunschärfe-Effekt, 45° Neigung, 15 px Verschiebung
2. Bewegungsunschärfe-Effekt, 45° Neigung, 30 px Verschiebung
3. Rausch-Effekt, Gauss-Verteilung, 15% Rauschen, nicht monochrom
4. Rausch-Effekt, Gauss-Verteilung, 30% Rauschen, nicht monochrom
5. Kontrast- und Helligkeitsverringerung, 70% weniger Kontrast, 70% weniger Helligkeit
6. Verzerrung, Säuleneffekt, 40%
7. Verzerrung, Säuleneffekt, 80%
8. Verkleinerung, 50% Größenreduktion
9. Verkleinerung, 75% Größenreduktion
10. Leichte perspektivische Verzerrung
11. Starke perspektivische Verzerrung

Beim Testen der Algorithmen legte die Projektgruppe auf folgende Faktoren wert:

1. Gefundene Keypoints pro Bild
2. Durchschnittliche Keypoints über das Testset hinweg
3. Falsch-Positiv, das bedeutet falscherkannte Übereinstimmungen bei Bildern
4. Geschwindigkeit
5. Anzahl der Fehlschläge

Um die Algorithmen zu testen, wurden einheitliche Schnittstellen benötigt, welche zum Zeitpunkt der zweiten Iteration von Release II noch festzulegen waren. Aus diesem Grund konnte eine entgültige Implementierung der Testumgebung in der Iteration noch nicht erfolgen.

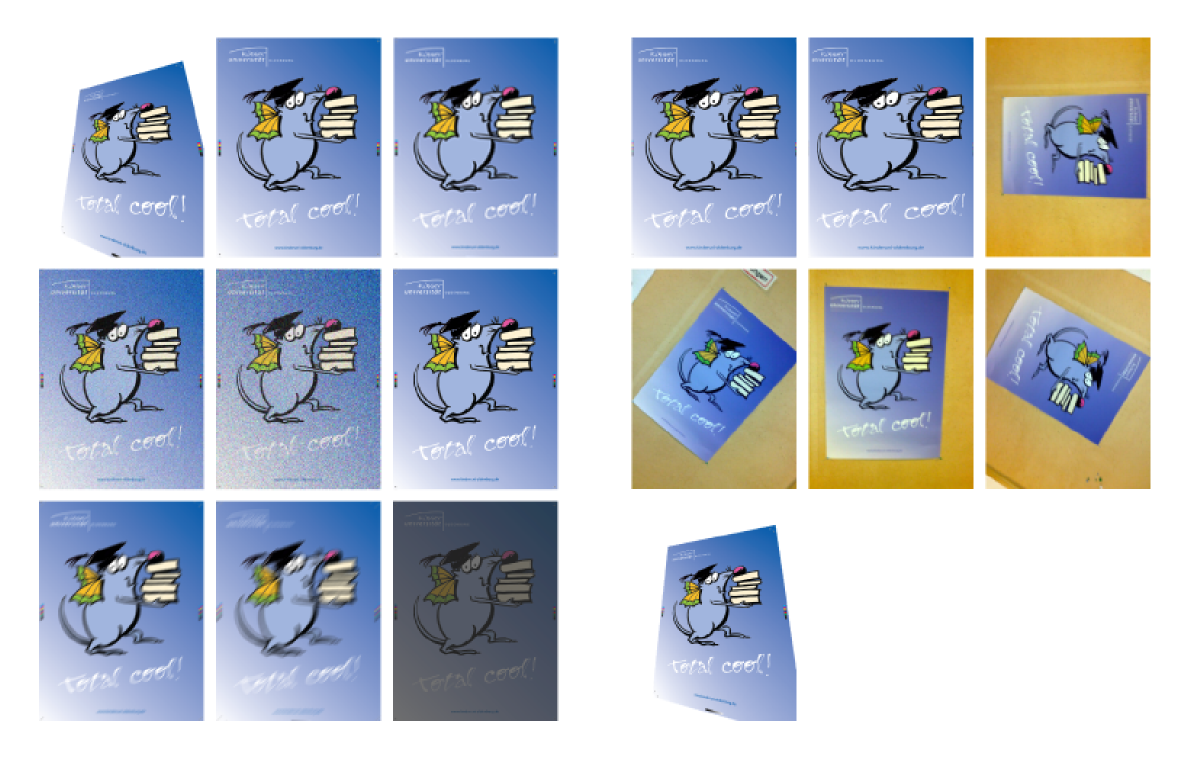


Abbildung 9.4.: Beispiel eines Bilder-Testsets

Evaluation mit Menschen

Als Termin für eine Evaluation mit Menschen wurde der 20.03.08 festgelegt, an dem der Leitfaden, der Ablauf der Evaluation, der Fragebogen an die Testpersonen, ein Flyer zum Austeilen sowie die Testplakate fertiggestellt waren.

Die Grobplanung sah eine Art Stand an einem bestimmten Ort wie zum Beispiel dem Mensa-Foyer oder auch der Innenstadt vor, an dem eine Live-Demo stattfinden sollte. An dem Stand sollten dann von Testpersonen, anhand einer Aufgabenstellung, die dort hängenden, von der Projektgruppe gestellten, Plakate mit dem Handy abfotografiert werden, um die Funktionalität der Software zu demonstrieren. Anschließend sollten die Testperson einen aus Multiple-Choice-Fragen, sowie Freitext bestehenden Fragebogen ausfüllen, welcher zur Evaluation ausgewertet wurde.

Besonderes Augenmerk lag bei der Evaluation auf:

1. Usability
2. Anwendungsverständnis
3. Featurepräferenz (Was gefällt, was fehlt)
4. Zielgruppe (Alter, Geschlecht)
5. Selbsteinschätzung (Kenntnisse mit mobilen Entgeräten)

6. Ergonomie
7. ggf. „Wizard-of-Oz“-Test nicht vorhandener Features
8. ggf. Webdemonstration mit anschließenden Fragebogen

Für die Umsetzung mussten folgende Rahmenbedingungen geschaffen werden:

- Publicity für weiterführende Informationen (z.B.: Flyer, Visitenkarten)
- Bedienungsanleitung für eine einheitliche Einführung
- Zusammenstellung von Plakaten
- Ablaufplan für die Durchführung
- Fragebögen

Besonderheiten bei der Implementierung

Aus der Evaluation (Abschnitt 8.4.2) folgte, dass der SIFT-Algorithmus und der Brute-Force-Ähnlichkeitsvergleich angemessene Qualität bieten. Daher wurde weiterer Aufwand in die Java-Portierung der OpenCV-Variante investiert. Parallel wurde eine Java-Implementierung[2] gefunden, die u.a. auf ImageJ[3] basiert. Da es in der Java-Portierung noch Fehler gab und diese nicht in angemessener Zeit gefunden werden konnten, wurde die ImageJ-Variante in das Projekt integriert.

Um einen Vergleich auf den Referenzbildern zu beschleunigen, wurden die Nachbarn im K-D-Baum heuristisch ermittelt und nur einmal beim Serverstart geladen. Für den k-D-Baum wurde nun eine vorhandene Implementierung[4] genutzt.

Systemarchitektur

In dieser Iteration fanden einige Erweiterungen der bereits bestehenden Systemarchitektur statt. Ein Klassendiagramm des Handy-Clients ist in Abbildung 9.5 dargestellt. Es wurde um die Klassen „Picture“ und „ImageTools“ erweitert. „Picture“ dient zur speichereffizienten Verwaltung von temporären Bildobjekten, da sie dem Entwurfsmuster „singleton“ folgt. Die Klasse „ImageTools“ wurde eingeführt, um Operationen zur Bildmanipulation auf der Clientseite zu kapseln. Zum Abschluss der Iteration enthielt sie die Funktionalität zur Skalierung von Bildern zwecks Anzeige innerhalb des Handy-Clients.

Einige aufwändigere Veränderungen fanden hingegen in der Server-Anwendung (vgl. 9.6) statt. Die Paketstruktur wurde um die Pakete „olivanda_server“ zur Kapselung

[2] http://fly.mpi-cbg.de/ saalfeld/javasift.html
[3] http://rsb.info.nih.gov/ij/
[4] http://www.savarese.com/software/index.html

aller Basisklassen, sowie „datamanagement“ zur Kapselung von Datenbankfunktionalitäten erweitert. Über das definierte Interface „DBInterface“ können Anfragen an eine austauschbare Datenbank gestellt werden. Diese wurde zunächst unter „DBConnection“ mit Mockup-Daten implementiert und liefert statische XML-Antworten zu angefragten Posterdaten.

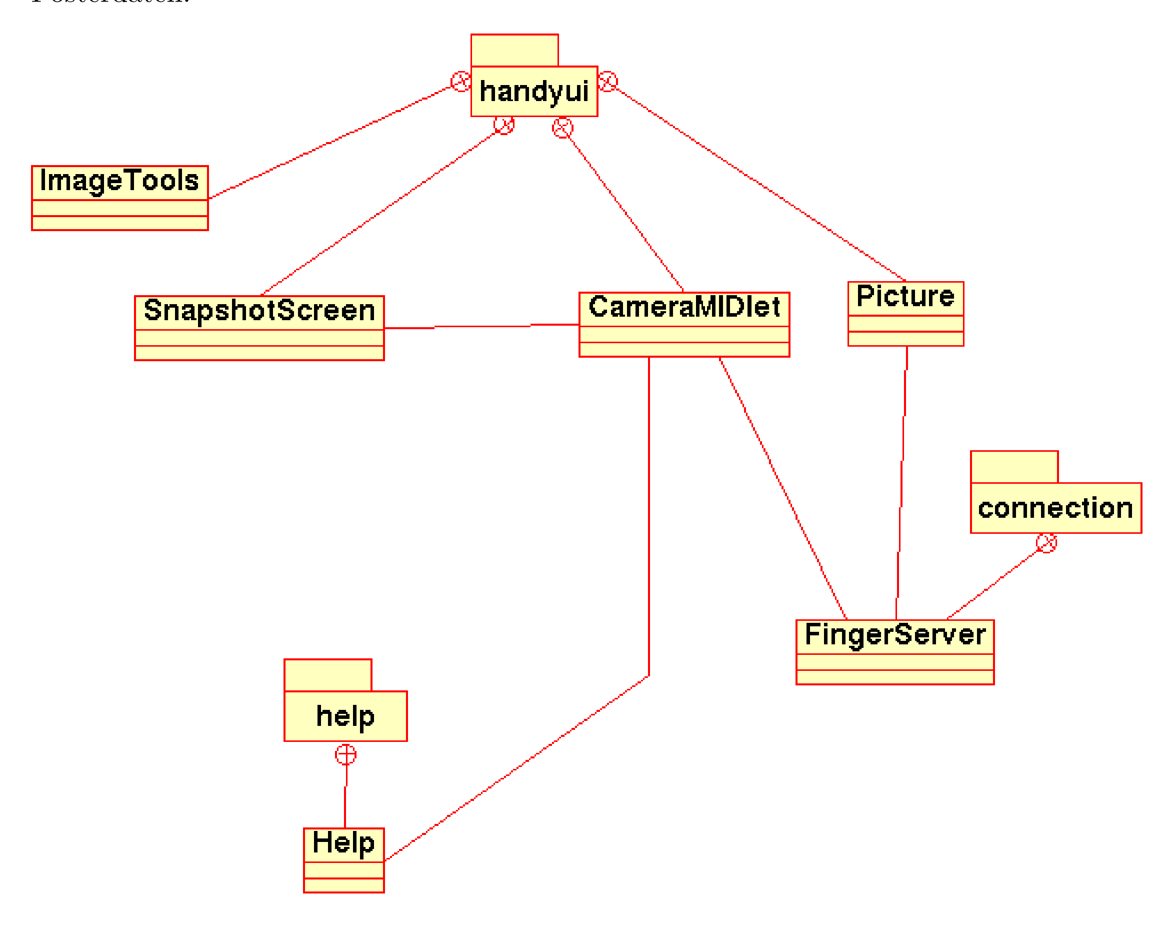


Abbildung 9.5.: Klassendiagramm des Handy-Clients

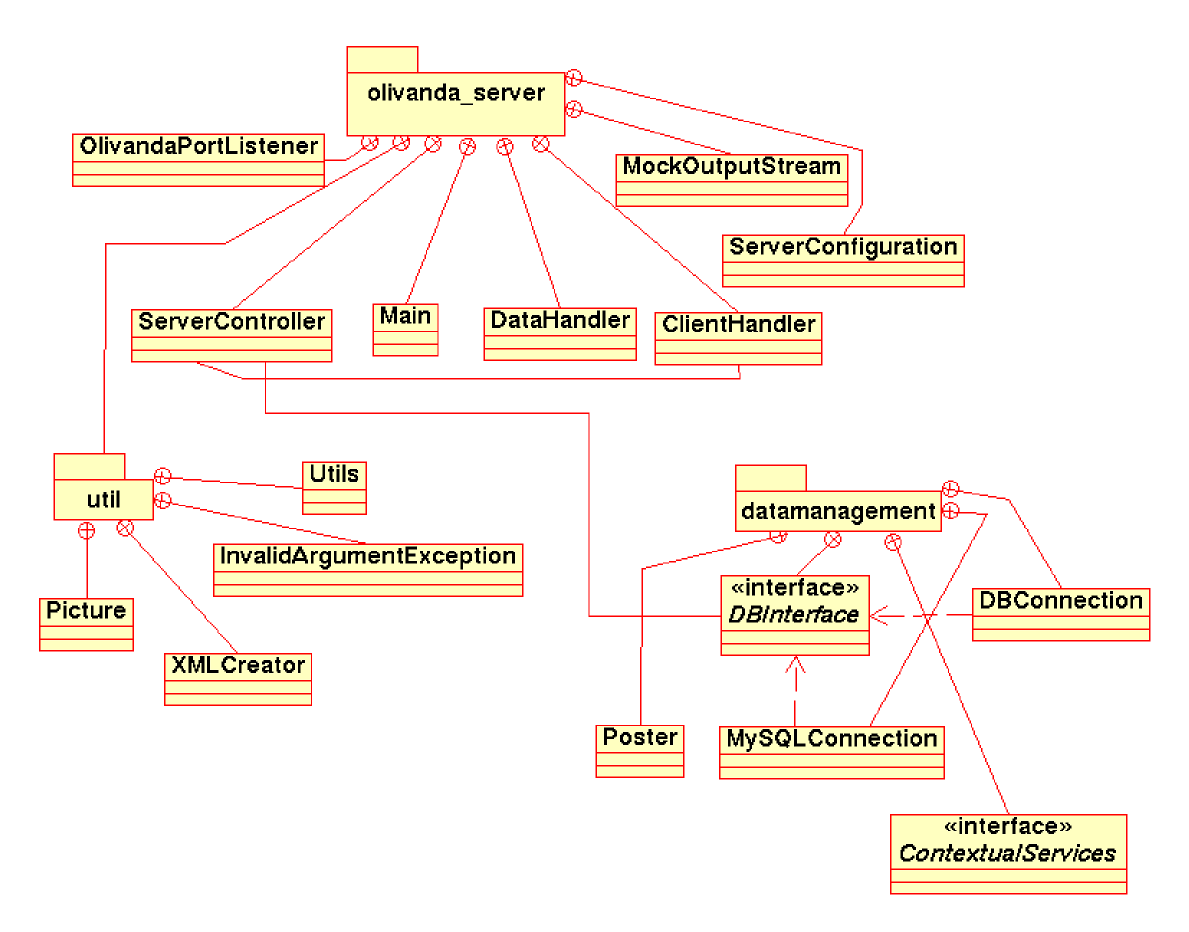


Abbildung 9.6.: Klassendiagramm des Servers

Client-Server-Kommunikation

Die Client-Server-Kommunikation war anfangs lediglich auf das Versenden von einzelnen Bildern ausgelegt worden. Dazu wurde zunächst die Länge des zu sendenden Bildes und anschließend die Bilddaten selbst als Bytestrom über eine Socket-Verbindung übertragen.

In dieser Iteration war es nötig, die Kommunikation dahingehend anzupassen, dass zusätzlich zu Bilddaten ebenfalls String-Daten als Vorbereitung für die Übertragung von kontextuellen Bookmarks zwischen Client und Server ausgetauscht werden können (vgl. 9.7).

Als Alternative zur Erweiterung der bestehenden Struktur wurde eine grundlegende Umgestaltung der Kommunikation auf Basis von XML-RPC[5] in Betracht gezogen, da in der genutzten J2ME Polish-API bereits clientseitige Funktionalitäten[6] hierfür angeboten werden.

[5]http://www.xmlrpc.com/
[6]http://www.j2mepolish.org/cms/leftsection/documentation/rmi-xml-rpc.html

Aufbau des Bytestroms zum Ende von Release II, Iteration I:

Bildlänge	Bilddaten

Anpassung in Release II, Iteration II:

Bildlänge	Bilddaten	Stringlänge	Stringdaten

Abbildung 9.7.: Anpassung der Client-Server-Kommunikation

Hierbei handelt es sich um eine XML-basierte Lösung für Remote-Prozeduraufrufe. Da als Datenkommunikationsschicht innerhalb des Systems das HTTP-Protokoll verwendet wurde, wäre hierfür eine grundlegende Anpassung der existierenden Serverarchitektur nötig geworden. Dies hätte erlaubt, Funktionsaufrufe auf Server- und Clientseite direkt durchzuführen und sowohl Bilder, String-Daten, als auch beliebige andere Objekttypen zu versenden.

Allerdings waren für das Projekt keine entscheidenden Vorteile ersichtlich, da die bereits existierende Struktur stabil funktionierte und abgesehen von der fehlenden Möglichkeit zur Übertragung von String-Daten eine ausreichende Funktionalität bot. Eine aufwändige Umstellung hätte somit dem „simplicity"-Grundsatz des XP widersprochen.

Als weiteres Gegenargument für die Verwendung von XML-RPC sprach die für den Verlauf des Projekts eventuell angepeilte Verwendung der MMS-Technologie als Basis für den Datenaustausch. Eine Umstellung auf MMS-Technologien wurde allerdings aus Kosten- und ebenfalls „simplicity"-Gründen zunächst nicht geplant. Es wurde jedoch festgehalten, in Zukunft bei weiteren Anpassungen der Client-Server-Struktur erneut die Verwendung von XML-RPC bzw. MMS zu prüfen.

Somit wurde in Iteration II die Client-Server-Kommunikation dahingehend angepasst, dass über die Socketverbindung neben Bilddaten ebenfalls String-Daten übertragen werden konnte (vgl. 9.7).

Die nächste Designentscheidung bestand darin, die zu verwendende Struktur für die String-Daten zu definieren. Hierfür standen die Technologien JSON[7] sowie XML[8] zur Auswahl. Die Wahl fiel auf XML, da es sich bei JSON in erster Linie um ein Format zum Datenaustausch handelt.

Für die Darstellung von Serverantworten innerhalb des Handy-Clients waren jedoch Features wie Textformatierung und Textstrukturierung notwendig, die sich durch die Verwendung von XML leicht realisieren ließen. Hierdurch trat der Vorteil der schlankeren Kodierung von JSON gegenüber XML in den Hintergrund (vgl. [10]).

[7]http://json.org

[8]http://www.w3.org/XML/

Zur Erstellung und zum Auslesung von XML-Daten auf der Serverseite wird die jdom-Bibliothek[9] verwendet. Die J2ME Polish-API stellt bereits ohne zusätzliche Erweiterungen Funktionalitäten zur Erstellung und zum Auslesen von XML-Daten zu Verfügung.

Da das Konzept zur Gestaltung von kontextuellen Bookmarks sich noch in der Entwicklung befand, wurde zunächst ein einfaches XML-Dokument mit zugehörigem XML-Schema definiert, das lediglich zur Anzeige eines Plakattitels als Antwort des Servers auf eine Client-Anfrage innerhalb des Clients diente. Im nachfolgenden ist ein XML-Dokument mit dem dazugehörigen Schema beispielhaft aufgeführt, wie es nach Iteration II auch in der Implementierung zum Einsatz kam:

```
XML-Dokument:

<?xml version="1.0" encoding="UTF-8"?>
<olivanda:bookmark  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
   xmlns:olivanda='http://xml.netbeans.org/schema/ProtocolSchema'>
    <olivanda:title>Mein Plakattitel</olivanda:title>
    <olivanda:thumbnail>plakat.jpg</olivanda:thumbnail>
</olivanda:bookmark>

XML-Schema:

<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            targetNamespace="http://xml.netbeans.org/schema/ProtocolSchema"
            xmlns:tns="http://xml.netbeans.org/schema/ProtocolSchema"
            elementFormDefault="qualified">

<xsd:element name="bookmark">
    <xsd:element name="thumbnail" type="xsd:string" maxOccurs="1"/>
    <xsd:element name="title" type="xsd:string" maxOccurs="1"/>
</xsd:element>

</xsd:schema>
```

9.3.3. Ergebnisse

Es wurde ein konkretes Konzept zur Evaluation der Software mit Menschen erarbeitet. Der Handy-Client wurde mit Hinblick auf diese Evaluation angepasst. Erste Ideen in Bezug auf die Einbindung kontextueller Bookmarks wurden gesammelt; die Ausarbeitung eines konkreten Aufbaus dieser Bookmarks stand aber noch aus und wurde für Iteration III angesetzt.

[9] http://www.jdom.org

Ergebnisse der Akzeptanztests

Die Akzeptanztests verliefen weitgehend zufriedenstellend. Ein fotografiertes Plakat, das nicht in den Referenzbildern vorhanden war, wurde jedoch fälschlicherweise als ein Referenzbild aus dem Emulator erkannt. Die Ursache hierfür könnte zum einen an einem zu niedrigen Grenzwert für nicht erkannte Bilder liegen und zum anderen an der sehr geringen Auflösung (160 x 120 Pixel) der Referenzbilder aus dem Emulator. Im Vergleich zu den Emulatorbildern haben die anderen Referenzbilder in der Regel eine Größe von ca. 800 x 600 Pixeln.

Desweiteren konnte die Systemantwortzeit nicht zufriedenstellend erreicht werden, da der Server auf einem recht langsamen Notebook instantiiert war. Daraus ergab sich auch, dass der Akzeptanztest nicht optimal formuliert wurde, da keine konkrete Zielmaschine vorgegeben war.

Retrospektive

Diese Iteration war von kleineren Verbesserungen und Bugfixes in der Software und intensiver Arbeit am Endbericht geprägt. Von vielen Projektgruppenmitgliedern wurde der Aufwand zur Dokumentation als unangenehm hoch empfunden. Das Vorgehensmodell folge nicht mehr dem Prinzip „travel light“, was einen zentralen Punkt des XP-Modells darstellt.

Bei der Entwicklung des Handyclients ergaben sich in dieser Iteration erste Probleme bezüglich des Speicherverbrauchs und der Geschwindigkeit. Deren Behebung gestaltete sich schwierig, da sie am Emulator nicht reproduzierbar waren. Dennoch mussten sie behoben werden, um die Weiterentwicklung des Clients zur Darstellung kontextueller Bookmarks zu ermöglichen.

Gegen Ende der Iteration ergaben sich zudem organisatorische Probleme; so musste das wöchentliche Meeting im OFFIS in einer Woche verschoben werden, wodurch die Auswahl der neuen Storycards für die dritte Iteration erst drei Tage nach Beginn derselben stattfinden konnte. Zudem führten einige Krankheitsfälle und die etwas unvorteilhafte Urlaubsplanung der Gruppenmitglieder – im Mittel waren etwa fünf von neun anwesend – zu verringerter Arbeitskraft. Durch effektive Zusammenarbeit konnten dennoch alle zu bearbeitenden Storycards und Tasks umgesetzt werden.

In Bezug auf das XP stelle sich eine gewisse Nachlässigkeit ein, so dass anstelle von echtem Pair Programming teilweise Einzelarbeit mit nachfolgendem Review stattfand. Die evtl. im Einzelfall kurzfristig gewünschten Pausen vom Pair Programming werden in der Literatur (vgl. [23]) allerdings erwähnt und es war zu erwarten, dass sich diese kurzfristige Entwicklung bald revidieren würde.

9.4. Iteration III

Die dritte Iteration des zweiten Releases erstreckte sich über die Kalenderwochen 13 bis 19 und verlief im Übergang von der vorlesungsfreien Zeit zur Vorlesungszeit. Die Iteration wurde um drei Wochen verlängert, um die Evaluation des Systems zu planen und durchzuführen (siehe auch Unterkapitel 9.4.1).

In Abbildung 9.8 ist die Einbettung der Iteration in den Gesamtablauf dargestellt.

9.4.1. Zeitplan

Durch die Konzeption und Durchführung der Evaluation, sowie die damit verbundene Aussetzung der Entwicklung musste der Zeitplan etwas verändert werden. Die Iteration III wurde um drei Wochen verlängert. Sie dauerte damit sieben statt wie geplant vier Wochen. Die nachfolgenden Releases mussten um drei Wochen verschoben werden. Das Release IV wurde auf zwei Iterationen reduziert, da bis zum – fest auf den 30. September 2008 terminierten – Projektende nicht mehr genug Zeit für weitere Iterationen sein würde. Der ursprüngliche Zeitplan (Abb. 9.1) wurde durch den Nebenstehenden ersetzt.
Es wurde entschieden, das Konzept der vierwöchigen Iterationen beizubehalten und die Anzahl der Iterationen zu reduzieren, statt gleichbleibend viele, jedoch kürzere Iterationen durchzuführen, da die vierwöchigen Iterationen allgemein als effizienter empfunden wurden.

Kalenderwoche	Release	Iteration	Woche
(01.01.2008)	1	1	1
3	1	2	1
4	1	3	1
5	1	4	1
(04.02.2008)	2	1	1
7	2	1	2
8	2	1	3
9	2	2	1
10	2	2	2
11	2	2	3
12	2	2	4
13	2	3	1
14	2	3	2
15	2	3	3
16	2	3	4
17	2	3	5
18	2	3	6
19	2	3	7
(12.05.2008)	3	1	1
21	3	1	2
22	3	1	3
23	3	1	4
24	3	2	1
25	3	2	2
26	3	2	3
27	3	2	4
28	3	3	1
29	3	3	2
30	3	3	3
31	3	3	4
(04.08.2008)	4	1	1
33	4	1	2
34	4	1	3
35	4	1	4
36	4	2	1
37	4	2	2
38	4	2	3
(30.09.2008)	4	2	4

Abbildung 9.8.: Aktualisierte Zeitplanung Release III

9.4.2. Storycards

In Iteration 3 des Releases 2 wurden die Storycards „Projekt-Dokumentation“, „Evaluation der Algorithmen“, „Evaluation mit Menschen“, „Client GUI anpassen“, „Aufbau kontextueller Bookmarks“, „Dienste in Bookmarks“, „Website aktualisieren“ und

„Konzept zur Personalisierung“ verwendet. Für den Akzeptanztest der jeweiligen Storycard galten die bereits in der letzten Iteration angegebenen allgemeinen Vorbedingungen (vgl. Abschnitt 9.3.1). Erneut wurde auf einen Akzeptanztest der Storycard „Projekt-Dokumentation“ verzichtet.

Projekt-Dokumentation

Lfd-Nummer: 1		
Projekt-Dokumentation		
Beginn: 01.10.2007	Risiko: niedrig	Zeit: bis Projektende
BESCHREIBUNG		
Die Dokumentation des Projektes wird durch die Verwendung von XP iterativ ergänzt und nach Abschluss des Projektes müssen die verschiedenen Aspekte der Projektarbeit in einer Dokumentation nachlesbar sein. Diese wird online einsehbar sein und die Projekt-Homepage bietet einen aktuellen und optisch ansprechenden Überblick über den Status der Projektarbeit. Der Projekt-Endbericht wird fortlaufend iterativ erweitert.		
TASKS	VERANTWORTLICH	ERLEDIGT
Klassendiagramm als rückblickende Übersicht	Jan	X
Grundgerüst des Endberichts	Christian M.	X
Benutzerhandbuch		
Online-Hilfe		
Umstrukturierung des Endberichts	Marcel	X
Erfassung der Daten, die auf die Website gehören	Dennis	X

Evaluation der Algorithmen

Lfd-Nummer: 8		
Evaluation der Algorithmen		
Beginn: 19.02.2008	Risiko: niedrig	Zeit: 60 h

BESCHREIBUNG
Die Algorithmen werden anhand von verschiedenen Gesichtspunkten, wie Geschwindigkeit, Treffergenauigkeit und Robustheit einheitlich getestet und bewertet. Dies geschieht mit Hilfe eines festen Testmusters, welches ein Bilderset und fixe Grenzwerte beinhaltet.

Vorbedingung:

- Keine Vorbedingungen.

Nachbedingung:

- Der Server hat eine Ausgabe der zur Identifikation eines Testbildes benötigten Zeit, dem dazu erkannten Referenzbild und die Anzahl an übereinstimmenden SIFT-Keypoints ausgegeben.

TASKS	VERANTWORTLICH	ERLEDIGT
Überlegen eines Grobkonzeptes für die Evaluation	Marcel & Sören	X
Refactoring des Konzepts mit dem Betreuer	Marcel & Sören	X
Referanzplakate organisieren	Marcel & Sören	X
Referenzbilder aufbereiten	Marcel & Sören	X
Testsets zusammenstellen	Marcel & Sören	X
Algorithmentest implementieren	Marcel & Sören	X
Auswertung erstellen	Marcel & Sören	
Evaluation der Aufnahmeauflösung		
RMI/RPC recherchieren		
JSon evaluieren bzgl. Geschwindigkeit und Unterstützung durch J2ME-Polish		

Evaluation mit Menschen

Lfd-Nummer: 9	Priorität:	
Evaluation mit Menschen		
Beginn: 19.02.2008	Risiko: niedrig	Zeit: 120 h

BESCHREIBUNG
Die Entwickler einer Anwendung haben meist nur eine eingeschränkte und starre Sicht auf das von ihnen entwickelte Produkt und die Akzeptanz des Endkunden nach der Fertigstellung fehlt. Um eine möglichst hohe Akzeptanz unserer Software durch den Benutzer zu gewährleisten, muss die bestehende Anwendung mit Hilfe von real existierenden Personen evaluiert werden. Dadurch werden Defizite in der Bedienung noch während der Entwicklungsphase aufgedeckt und weiterer Entwicklungsbedarf des Projektes ermittelt. Hierzu wird zunächst ein Konzept entwickelt, das u.a. die zu befragende Zielgruppe, konkrete Fragebögen, sowie evtl. nötige Erweiterungen des Handy-Clients beeinhaltet.

Vorbedingung:

- Keine Vorbedingungen.

Nachbedingung:

- Der Handy-Client ist durch provisorische Menüs zur Darstellung von kontextuellen Bookmarks erweitert worden.
- Die Evaluation wurde durchgeführt und die Ergebnisse in einer Präsentation vorgestellt.

TASKS	VERANTWORTLICH	ERLEDIGT
Überlegen eines Grobkonzeptes für die Evaluation	Marcel & Sören	X
Refactoring des Konzepts mit dem Betreuer	Marcel & Sören	X
Produktion eines Flyers, um die Ziele der PG für externe Personen darzustellen	Sören	X
Dummy-Bookmarks einfügen	Marcel, Sören & Torben	X
Menü-Engine, die Menüpunkte für Bookmarks dynamisch erzeugt		
Fragebogen überarbeiten (5-Punkt-Skala)	Sören	X
Absprache mit Betreuern	Sören, Marcel	X

Client GUI anpassen

Lfd-Nummer: 10	Priorität:	
Client GUI anpassen		
Beginn: 04.03.2008	Risiko: hoch	Zeit: 50 h

Beschreibung
Die Attraktivität einer Anwendung für einen Benutzer steigt durch eine intuitive Bedienung, sowie einer optisch ansprechende Gestaltung. Durch ständige Erweiterungen und Strukturumstellungen muss der Handy-Client fortlaufend angepasst werden. Die grafische Oberfläche des Handy-Clients wird konsequent einen einheitlichen Gestaltungsstil verwenden und ansprechend gestaltet sein. Alle Elemente des UIs werden nach demselben Stil aufgebaut, gestaltet und bedienbar sein. Der optische Aufbau wird sich an anerkannte Standards des Software Engineerings halten. So können alle Funktionalitäten innerhalb von maximal drei bis vier Schritten erreicht werden.

Vorbedingung:

- Keine Vorbedingungen.

Nachbedingung:

- Der Handy-Client besitzt einen einheitlichen Gestaltungsstil.
- Der Entwurf für das grafische Layout des Clients wurde präsentiert.
- Alle Anwendungsfunktionen sind in maximal drei bis vier Schritten erreichbar.

Tasks	Verantwortlich	Erledigt
Ladebalken umgestalten (keine Nummern mehr anzeigen, „Sanduhr“)	Christian M. & Christian N.	X
Einen Entwurf zur grafischen Gestaltung des Handy-Clients erstellen	Jeschua	X
Die Elemente der Handy-GUI auf eine einheitliche Gestaltung überprüfen	Jeschua	X
Die Elemente der Handy-GUI an eine einheitlich Gestaltung anpassen		
Skalierung der aufgenommenen Fotos auf die Größe des verwendeten Handy-Displays	Christian M. & Christian N.	X
GUI verschnellern, Speicherverbrauch minimieren	alle	X

Lfd-Nummer: 10	Priorität:	
Client GUI anpassen (Forts.)		
TASKS	VERANTWORTLICH	ERLEDIGT
Refactoring allgemein und speziell CameraMIDlet	alle	X
J2ME-Polish auf neue Version updaten	Christian N., Jeschua	X
Hilfe-Bug reparieren	Jeschua, Jan	X
IP-Eingabe überarbeiten		
Bug (Thread-Exception) in „Bild versenden“ beheben (tritt auf, wenn man zurück will	Oliver, Jan	X

Aufbau kontextueller Bookmarks

Lfd-Nummer: 13	Priorität:	
Aufbau kontextueller Bookmarks		
Beginn: 04.03.2008	Risiko: hoch	Zeit: 40 h

Beschreibung
Um Hintergrundinformationen und weitere Dienste zu Plakaten zu erhalten, wird der Benutzer über das Versenden von gemachten Fotos kontextuelle Bookmarks erstellen und anfragen können, die ihm weiterführende und evtl. anschließend verfügbare Dienste zur Verfügung stellen. Die Problematik besteht darin, ein Konzept zur Gestaltung und Repräsentation von kontextuellen Bookmarks zu finden und ggf. zu erstellen. Hierzu wird zunächst nach bereits existierenden Lösungsansätzen recherchiert und darauf basierend ein Konzept entworfen, wie durch den Benutzer angefragte Bookmarks in sich aufgebaut sind und wie sie ihm präsentiert werden. Nach Erstellung des Konzeptes muss dieses in der Anwendung umgesetzt werden.

Vorbedingung:

- Keine Vorbedingungen.

Nachbedingung:

- Es wurde ein Konzept zum Aufbau kontextueller Bookmarks erstellt und in einer Präsentation vorgestellt.
- Die Umsetzung des Konzeptes wird mit Hilfe einer Präsentation und Demonstration mit Hilfe der Anwendung dargestellt.

Tasks	Verantwortlich	Erledigt
„State of the art“- Analyse bestehender Techniken	Christian M., Christian N.	X
Konzeptentwurf zum Aufbau eines Bookmarks	Jan, Oliver, Christian M.	X
Konzeptentwurf zur Repräsentation von Bookmarks auf dem Handy-Client	Dennis, Jan, Christian N.	X
Implementierung einer Datenstruktur für Bookmarks	Chr. M., Jeschua, Oliver, Torben	X
Laden/Speichern von kontextuellen Bookmarks auf dem Handy	Sören, Oliver	X

Dienste in Bookmarks

Lfd-Nummer: 14	Priorität:	
Dienste in Bookmarks		
Beginn: 04.03.2008	Risiko: hoch	Zeit: 80 h

Beschreibung
Dem Benutzer sollen neben einfachen Hintergrundinformationen zu fotografierten Plakaten auch weitere externe Dienste und Dienstleistungen angeboten werden. So müssen die vom Benutzer angefragten Bookmarks mit externen Diensten und Dienstleistungen verknüpft werden. Hierzu muss zunächst ein Lösungsansatz zur Einbettung von Diensten und Dienstleistungen entwickelt und testweise mit Mockup-Diensten implementiert werden. Der Benutzer wird so über die für ihn gespeicherten Bookmarks Zugriff auf dazu passende Dienste/Dienstleistungen erhalten.

Vorbedingung:

- Keine Vorbedingungen.

Nachbedingung:

- Es wurde ein Konzept zur Einbettung von Diensten und Dienstleistungen in kontextuelle Bookmarks entwickelt und in einer Präsentation vorgestellt.
- Die Implementierung von Mockup-Diensten wird mit Hilfe der Anwendung demonstriert.

Tasks	Verantwortlich	Erledigt
Konzept zur Einbettung von Diensten/Dienstleistungen in Bookmarks erstellen		
Realisierung des erstellten Konzeptes mit Mockup-Diensten	???	X
Anpassung und Erweiterung der Realisierung auf real existierende Dienste		
Konzeptentwurf Datenbank für Dienste	Christian M. & Dennis	X
Polish-Browser für Dienst-URLs ausprobieren	Jan, Sören, Oliver	

Website aktualisieren

Lfd-Nummer: 15	Priorität:	
Website aktualisieren		
Beginn: 04.03.2008	Risiko: niedrig	Zeit: bis Projektende

Beschreibung
Die Website eines Projektes dient als Aushängeschild und Repräsentation in der Öffentlichkeit und stellt den aktuellen Stand einer Projektarbeit dar. Durch die ständige Weiterentwicklung der Anwendung und damit verbundener Erreichung neuer Releases und Iterationen ist es notwendig, die Daten der Website aktuell zu halten. Die Website bietet den aktuellen Entwicklungsstand der Anwendung sowie eine Darstellung des Projektes (Beschreibung, Zielsetzung, Vorgehensweise und aktuellen Entwicklungsstand) selbst.

Vorbedingung:

- Keine Vorbedingungen.

Nachbedingung:

- Die Webseite zeigt den aktuellen Stand der Entwicklung, eine Beschreibung und die Zielsetzung des Projektes.

Tasks	Verantwortlich	Erledigt
Erfassung der Daten, die auf die Website gehören	Dennis	
Stand des Projektes auf der Homepage grafisch darstellen	Dennis	

Konzept zur Personalisierung

Lfd-Nummer: 17	Priorität:	
Konzept zur Personalisierung		
Beginn: 31.3.2008	Risiko: hoch	Zeit: 50 h
Beschreibung		
Die dem Benutzer angebotenen Bookmarks sollen personalisiert werden, um die Akzeptanz für die angebotenen Dienste zu erhöhen. Verschiedene Verfahren zur Personalisierung stehen zur Verfügung und bieten Vor- und Nachteile. Es muss evaluiert werden, welche Personalisierungsverfahren in diesem Fall am besten geeignet sind und wie die Personalisierung umgesetzt werden kann.		
Vorbedingung: • keine Vorbedingungen. *Nachbedingung:* • Eine „state-of-the-art“ Analyse wurde durchgeführt, ein Verfahren wurde darauf basierend ausgewählt und in einer Präsentation vorgestellt.		
Tasks	Verantwortlich	Erledigt
Recherche möglicher Personalisierungsverfahren	Christian N. & Christian M.	X
Entscheiden, welche Verfahren geeignet sind		
Konzept zur Identifikation des Clients beim Server		
Konzept: Historie bisher aufgerufener Dienste für jeden Benutzer		

9.4.3. Umsetzung

Die Arbeiten in dieser Iteration wurden von den Vor- und Nachbereitungen für die Evaluation bestimmt. Teilweise mussten aufgrund der knappen zur Verfügung stehenden Zeit Kompromisse eingegangen werden, so dass Programmcode entstand, der nach der Evaluation wieder entfernt werden sollte. Die Konzipierung, Durchführung und Ergebnisse der Evaluation werden im Kapitel 4 vorgestellt.

Betrachtete Algorithmen

Es wurden in der Iteration algorithmisch keine wesentlichen Veränderungen vorgenommen. Die Evaluation des bereits eingesetzten SIFT-Algorithmus wurde fortgesetzt, wobei insbesondere getestet wurde, wie robust sich das Verfahren gegenüber Deformatio-

nen und anderen Beeinträchtigungen der Eingabebilder verhält. Es wurden jedoch keine Veränderungen am eigentlichen Verfahren vorgenommen.

Zur Übertragung und Speicherung der kontextuellen Bookmarks wurden verschiedene Möglichkeiten in Betracht gezogen, aufgrund der auch clientseitig guten Unterstützung bot sich jedoch eine Übertragung und Speicherung als XML besonders an, so dass diese Methode gewählt wurde.

Besonderheiten bei der Implementierung

Für die Erzeugung des zu übertragenden XMLs wurde serverseitig ein Template verwendet, das auch die Struktur der XML-Repräsentation vorgibt:

```
<?xml version="1.0" encoding="UTF-8"?>
<olivanda:bookmark  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
   xmlns:olivanda='http://xml.netbeans.org/schema/ProtocolSchema'>

    <olivanda:title></olivanda:title>
    <olivanda:thumbnail></olivanda:thumbnail>
    <olivanda:service>
        <olivanda:attribute name="attr_name"></olivanda:attribute>
    </olivanda:service>

</olivanda:bookmark>
```

Dies ermöglicht die Übertragung eines Bookmarktitels und Thumbnails – welches Base64 kodiert übertragen werden könnte, falls es sich um ein Binärformat wie JPEG oder PNG handelt – sowie eines Zusatzdienstes.

Leider ist bei der Verarbeitung des Templates in Java keine Möglichkeit gegeben, Attribute zu wiederholen, so dass die Einbettung beliebig vieler Dienste nicht gelang. Dies wurde aber als unkritisch angesehen, da für die Auswahl der Mockup-Bookmarks der Evaluation bereits der Bookmarktitel bzw. eine dort eingebettete ID ausreichend war. Eine Überarbeitung der XML-Erzeugung wurde für die darauffolgende Iteration angesetzt.

Tatsächlich wurden die für die Evaluation benötigten drei Bookmarks fest im Client kodiert und lediglich anhand des Übertragenen XMLs bestimmt, welcher Bookmark angezeigt werden sollte. Auch die Erzeugung der Bookmark-Anzeige im Client wurde für die Evaluation fest einkodiert, um Zeit zu sparen. Für die Auswahl und den Aufbau der in der Evaluation verwendeten Mockup-Bookmarks siehe Abschnitt 4.1.

Systemarchitektur

Nach Abschluß der Storycards wurde eine Umstrukturierung der Clientarchitektur notwendig. Um Performance zu gewinnen und die Nachvollziehbarkeit des Codes sowie eine

bessere Grundlage zur Auffindung von Programmfehlern zu bieten, wurden alle grafischen Menüs in die Klasse „Menu“ ausgelagert (vgl. Klassendiagramm 9.9). Desweiteren wurde das Paket „beans“ eingefügt, um ein Datenmodell für die Repräsentation von kontextuellen Bookmarks zur Verfügung zu stellen. Um eine lokale Speicherung von diesen Bookmarks auf dem mobilen Endgerät zu ermöglichen, wurde die Klasse „Storage“ im Paket „utils“ hinzugefügt.

Weniger gravierend waren die Veränderungen an der Serverarchitektur (vgl. Diagramm 9.10). Das Paket „datamanagement“ wurde, wie das Paket „bean“ im Client, ebenfalls um ein Datenmodell zur Datenhaltung von kontextuellen Bookmarks erweitert.

Als weitere nennenswerte Änderung wurde das Interface „DBInterface“ von der Klasse „MySqlConnection“ implementiert, wodurch eine Anbindung zu einer MySQL Datenbank mit Plakat- und Kundendaten möglich wurde.

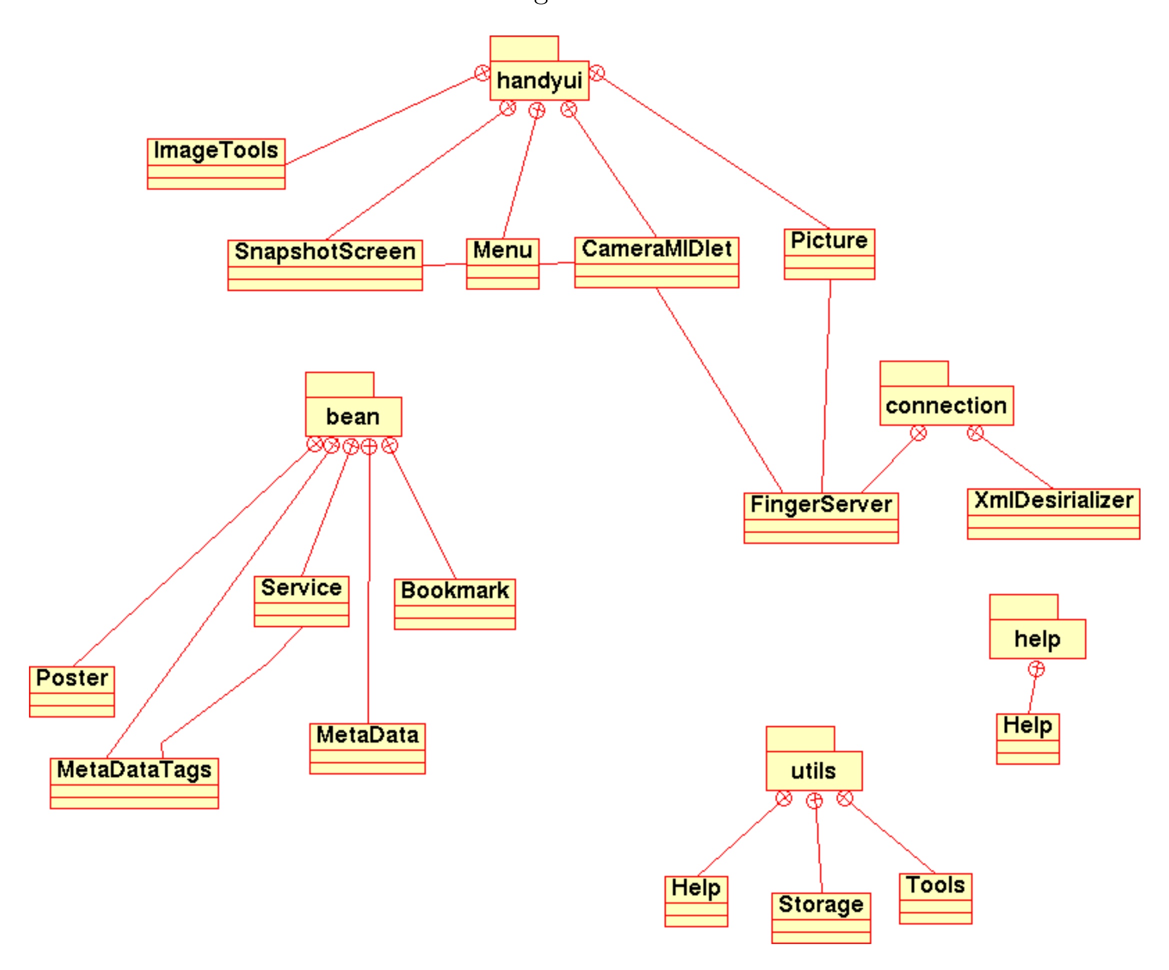


Abbildung 9.9.: Klassendiagramm des Handy-Clients

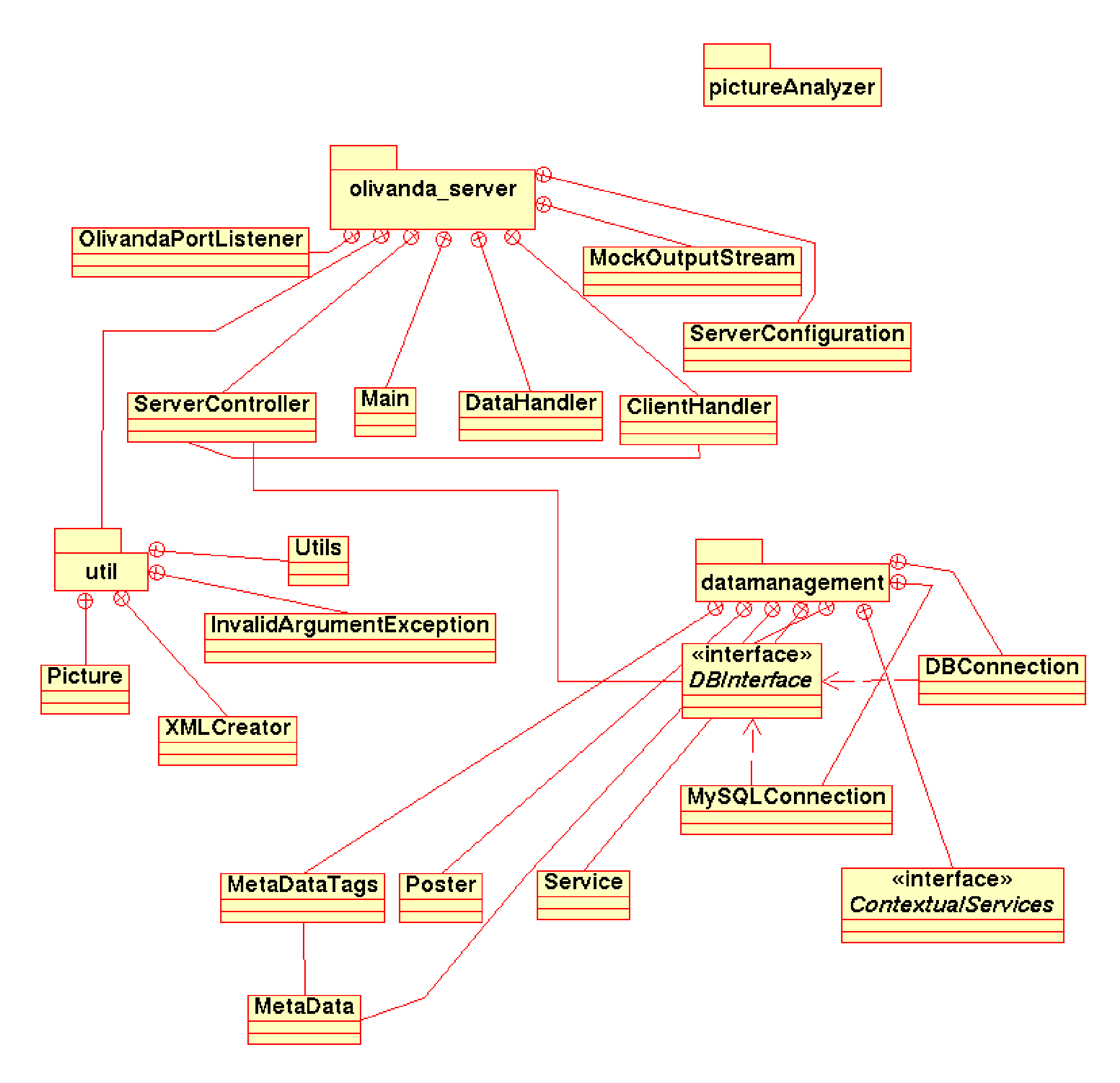


Abbildung 9.10.: Klassendiagramm des Servers

9.4.4. Ergebnisse

In der dritten Iteration des zweiten Releases entwickelten sich sowohl der Client als auch der Server zu einem stabilen System. Mit dem Client wurde es aufgrund eines Bugfixes endlich möglich, beliebig viele Fotos zu machen, ohne dass steigender Speicherverbrauch die Navigation im Menü stetig verlangsamt. Auch wurde die Möglichkeit kontextuelle Bookmarks auszulesen im Client implementiert.

Der Server lief zum Abschluss der Iteration sehr stabil und wurde auf die Erstellung von kontextuellen Bookmarks in XML vorbereitet. Allerdings war die Verarbeitung der

Bookmarks noch auf Dummybookmarks beschränkt. Dennoch funktionierten die Dummybookmarks so gut, dass zum Abschluss des Releases eine Evaluation der Software durchgeführt werden konnte.

Nach anfänglichen Startschwierigkeiten verlief diese äußerst erfolgreich, insbesondere nahmen weit mehr Personen teil als erwartet. Aufgrund der Evaluation und der Verlegung der wöchentlichen Treffen auf Donnerstags verschob sich jedoch der Gesamtzeitplan um drei Wochen. Dies führte dazu, dass das letzte Release am Projektende im Vergleich zu den vorangegangenen Releases um drei Wochen kürzer sein würde.

Für den Beginn des neuen Releases waren Refactoring-Maßnahmen innerhalb des Repositorys angedacht, da mehrere Gruppen an unterschiedlichen Bugs innerhalb derselben Klassen gearbeitet hatten, was zur Bildung mehrerer, sich parallel entwickelnder Versionen des Clients geführt hatte. Diese mussten kurz vor der Evaluation wieder zusammengeführt werden. Danach mussten die nun überflüssigen Nebenversionen des Clients wieder aus dem Repository entfernt werden.

Retrospektive

Fazit der Iteration war, dass alle Storycards und Tasks immer zeitig erfüllt wurden und auch das Zusammengehörigkeitsgefühl innerhalb der Gruppe, vor allem bestärkt durch Treffen in der Stadt und bei Grillabenden, immer stärker wurde.

Negativ wurde das Simplicity-Prinzip des XP empfunden. Zwar wird von allen Teilnehmern die Annahme gestützt, dass aufgrund dieses Prinzips immer nur das gerade Benötigte umgesetzt werden soll, um so keine unnötig komplexen Strukturen im Projekt zu bekommen, allerdings zeigte sich auch, dass nach jeder Iteration stets umfangreichere Refactoring-Maßnahmen erforderlich waren. Diese wären vermeidbar gewesen, wenn bereits durch eine längerfristige Planung der zukünftig benötigten Funktionalitäten bestimmte, zu dem Zeitpunkt noch nicht benötigte Strukturen im Projekt bei der Implementierung berücksichtigt worden wären.

10. Release III (Zeitraum: 19.05.2008 bis 03.08.2008)

Nachdem in den vorangegangenen Releases die grundlegende Funktionalität wie die Plakaterkennung und Auslieferung von Kontextuellen Bookmarks weitgehend fertiggestellt wurde, wurde nun auf diese Techniken aufbauend das System verfeinert und um zusätzliche Bausteine, die zum produktiven Betrieb nötig sind, erweitert.

Diese Schritte waren wichtig insbesondere deswegen, weil das dritte Release entsprechend des Zeitplans das letzte „große Release“ darstellen sollte; nach seinem Abschluss am 10. August war nur noch etwa ein Monat Zeit bis zum regulären Ende der Projektgruppe. Das für Anfang August angesetzte vierte Release konnte aufgrund der durch die Evaluation entstandenen Verzögerungen nur noch eine einzelne vollständige Iteration enthalten.

Somit sollten im dritten Release alle Arbeiten durchgeführt werden, die im Kontext des Gesamtprojektes für einen erfolgreichen Einsatz nötig sein würden, so dass nach dem Release lediglich Abschlussarbeiten wie das beseitigen von Fehlern in der Software, kleinere Anpassungen der Software und die Dokumentation bleiben würden.

10.1. Zielsetzung

Neben der bereits in den vorigen Releases angedachten und teilweise ausgearbeiteten Personalisierung bzw. Parametrisierung der Bookmarks sollte eine Möglichkeit geschaffen werden, die im System enthaltenen Plakate und Zusatzinformationen zu pflegen. Dies wurde in den vorangegangenen Releases nicht beachtet und musste nun von Anfang an konzipiert und umgesetzt werden. Im Gegensatz dazu wurde mit der Personalisierung bereits im zweiten Release begonnen und diese Arbeiten sollten fortgesetzt, aber auf konzeptioneller Ebene auch kritisch überdacht werden.

10.2. Zeitplan

Der Stand innerhalb des globalen Projektzeitplanes (vgl. 1.6) zu Beginn von Release III wird in Abb. 10.1 durch blau und lila hinterlegte Zeilen dargestellt und wurde im Vergleich zu Release II in einigen Punkten angepasst.
Aufgrund der durch die Evaluation in Release II entstandenen Verzögerungen wurde vor Beginn des dritten Releases eine Brückenwoche eingefügt, so dass das Release am 19. Mai statt – wie ursprünglich geplant – am 12. Mai begann:

- Iteration I:
  Start: 19.05.2008

- Iteration II:
  Start: 16.06.2008

- Iteration III:
  Start: 14.07.2008

Diese Änderung wirkte sich auch auf das nachfolgende vierte Release aus, was bedeutete, dass die globale Zeitplanung angepasst werden musste.

Kalenderwoche	Release	Iteration	Woche
(01.01.2008)	1	1	1
3	1	2	1
4	1	3	1
5	1	4	1
(04.02.2008)	2	1	1
7	2	1	2
8	2	1	3
9	2	2	1
10	2	2	2
11	2	2	3
12	2	2	4
13	2	3	1
14	2	3	2
15	2	3	3
16	2	3	4
17	2	3	5
18	2	3	6
19	2	3	7
(12.05.2008)	3	1	1
21	3	1	2
22	3	1	3
23	3	1	4
24	3	2	1
25	3	2	2
26	3	2	3
27	3	2	4
28	3	3	1
29	3	3	2
30	3	3	3
31	3	3	4
(04.08.2008)	4	1	1
33	4	1	2
34	4	1	3
35	4	1	4
36	4	2	1
37	4	2	2
38	4	2	3
(30.09.2008)	4	2	4

Abbildung 10.1.: Zeitplanung Release III

10.3. Iteration I

Iteration I fand, wie aus Abbildung 10.1 zu entnehmen, in den Kalenderwochen 21-24 und somit vollständig während der Vorlesungszeit statt. Sie stand im Zeichen der Nachbereitung der Evaluation, was den Programmcode betraf und der Konzeption der weiteren Softwarebausteine. Es musste ein webbasiertes Frontend für die Pflege des Datenbestandes im Server geplant und vollständig umgesetzt werden. Die Personalisierung der Dienste wurde – zunächst auf einfache Art und Weise – umgesetzt.

10.3.1. Storycards

Die in Iteration I eingesetzten Storycards bestanden weitestgehend aus Erweiterungen der Storycards von Release II, Iteration III. Jedoch wurden einige neue Storycards hinzugefügt, etwa für das Frontend zur Pflege des Datenbestandes. Zur besseren Übersicht sind hier alle Storycards aufgeführt.

Projekt-Dokumentation

Lfd-Nummer: 1		
Projekt-Dokumentation		
Beginn: 01.10.2007	Risiko: niedrig	Zeit: bis Projektende
BESCHREIBUNG		
Die Dokumentation des Projektes wird durch die Verwendung von XP iterativ ergänzt und nach Abschluss des Projektes müssen die verschiedenen Aspekte der Projektarbeit in einer Dokumentation nachlesbar sein. Diese wird online einsehbar sein und die Projekt-Homepage bietet einen aktuellen und optisch ansprechenden Überblick über den Status der Projektarbeit. Der Projekt-Endbericht wird fortlaufend iterativ erweitert.		
TASKS	VERANTWORTLICH	ERLEDIGT
Klassendiagramm als rückblickende Übersicht	Jan	X
Grundgerüst des Endberichts	Christian M.	X
Benutzerhandbuch	-	-
Online-Hilfe	-	-
Umstrukturierung des Endberichts	Marcel	X
Erfassung der Daten, die auf die Website gehören	Dennis	X

Website aktualisieren

Lfd-Nummer: 15	Priorität:	
Website aktualisieren		
Beginn: 04.03.2008	Risiko: niedrig	Zeit: bis Projektende

BESCHREIBUNG
Die Website eines Projektes dient als Aushängeschild und Repräsentation in der Öffentlichkeit und stellt den aktuellen Stand einer Projektarbeit dar. Durch die ständige Weiterentwicklung der Anwendung und damit verbundener Erreichung neuer Releases und Iterationen ist es notwendig, die Daten der Website aktuell zu halten. Die Website bietet den aktuellen Entwicklungsstand der Anwendung sowie eine Darstellung des Projektes (Beschreibung, Zielsetzung, Vorgehensweise und aktuellen Entwicklungsstand) selbst.

Vorbedingung:

- Keine Vorbedingungen.

Nachbedingung:

- Die Webseite zeigt den aktuellen Stand der Entwicklung, eine Beschreibung und die Zielsetzung des Projektes.

TASKS	VERANTWORTLICH	ERLEDIGT
Erfassung der Daten, die auf die Website gehören	Dennis	-
Stand des Projektes auf der Homepage grafisch darstellen	Dennis	-

Alternative Bilderkennungsalgorithmen und andere Erkennungsarten

Lfd-Nummer: 18a	Priorität:	
Alternative Bilderkennungsalgorithmen und andere Erkennungsarten		
Beginn: 12.05.2008	Risiko:	Zeit: 82 h

Beschreibung
Während der Entwicklungszeit des Projektes Olivanda ist bisher auf eine Recherche nach alternativen Technologien zur Bilderkennung nach Auswahl des SIFT-Algorithmus verzichtet worden. Um weitere Eigenschaften in Fotos, wie Gesichter und Text, zu erkennen, werden weitere Realisierungsmöglichkeiten von vergleichbaren Projekten auf ihre Anwendbarkeit überprüft. Falls vielversprechende Ansätze gefunden werden, sind diese im Projekt anzuwenden.

Vorbedingung:

- Alternative Techniken wurden recherchiert und diskutiert.

Nachbedingung:

- Die Ergebnisse wurden in einer Präsentation vorgestellt.

Tasks	Verantw.	Erl.	S.G.	S.V.
Suche nach alternativen Techniken zu SIFT und Präsentation erstellen	Torben & Marcel & Jan	X	17	11,5
Recherche nach OCR-Technologien und Präsentation erstellen	Torben & Jan	X	19	15
Recherche nach Gesichtserkennungs-Technologien und Präsentation erstellen	Torben	X	19	18
Recherche nach vergleichbaren Anwendungen und Präsentation erstellen	Marcel & Jan	X	9	7
Gesamtvergleich der Alternativen mit eigenem Projekt und anderen SIFT-Lösungen	Torben & Marcel & Jan	X	9	11,5
Diskussion über die weitere Vorgehensweise	Alle	X	9	9

Textdetektion und Texterkennung implementieren

Lfd-Nummer: 18b	Priorität:	
Textdetektion und Texterkennung implementieren		
Beginn: 12.05.2008	Risiko:	Zeit: n.a.

BESCHREIBUNG
In SC 10.3.1 wurden verschiedene Alternativen und Ergänzungen zu der bereits im Projekt realisierten SIFT-Variante diskutiert. Bei der weiteren Vorgehensweise wurde beschlossen, Textdetektion und Erkennung in das Projekt zu integrieren.

Vorbedingung:

- Alternative Techniken wurden recherchiert und diskutiert.

Nachbedingung:

- Texterkennung und Detektion wurden implementiert und in einer Live-Demo vorgestellt.

TASKS	VERANTW.	ERL.	S.G.	S.V.
Auswahl des zu nutzenden Frameworks für die Textdetektion und Texterkennung	Torben & Jan	-	-	-
Implementierung von Textdetektion und Texterkennung im Projekt	Torben & Jan	-	-	-

Frontend für kontextuelle Bookmarks

Lfd-Nummer: 19	Priorität: NUMMER	
Frontend für kontextuelle Bookmarks		
Beginn: 12.05.2008	Risiko:	Zeit: 102 h

Beschreibung

Dem Kunden der Anwendung ist es möglich, seine Plakatdaten in das bestehende System einzupflegen. Daher wird eine grafisch ansprechende Anwendung entworfen und implementiert, in der alle Funktionen in drei bis vier Schritten erreicht werden können. Der Kunde kann reine Bilddaten, dazu passende Services und Metadaten für seine Plakatkampagne über das Frontend einstellen. Diese sind dann automatisch für jeden Client verfügbar.

Vorbedingung:

- Frontend muss Zugriff auf die Datenbank des Olivanda-Projektes besitzen

Nachbedingung:

- Ein Plakatdatensatz kann in das aktiv laufende System mit Hilfe des Frontends eingebunden werden. Hierzu zählen Bilddaten, kontextuelle Bookmarks sowie genutzte Dienste des Plakaterstellers.
- Die Ergebnisse werden in einer Präsentation und einer Live-Demo vorgestellt. Die durch das Frontend beispielhaft eingestellte Plakatkampagne wird über den Handy-Client durch Aufnahme des entsprechenden Plakates abgerufen und mit den entsprechenden Bookmarks dargestellt.

Tasks	Verantw.	Erl.	S.G.	S.V.
Recherche nach Realisierungsmöglichkeiten und bestehenden Technologien	-	X	7	4
Technologieauswahl	-	X	9	4
Konzept einer Datenstruktur	-	-	16	11
Konzept der Benutzeroberfläche	-	-	10	19
Implementierung der Datenstruktur	-	-	34	5
Implementierung des Frontends	-	-	26	27

Evaluation der Algorithmen

Lfd-Nummer: 8	Priorität:	
Evaluation der Algorithmen		
Beginn: 19.02.2008	Risiko: niedrig	Zeit: 19 h

BESCHREIBUNG
Die Algorithmen werden anhand von verschiedenen Gesichtspunkten, wie Geschwindigkeit, Treffergenauigkeit und Robustheit einheitlich getestet und bewertet. Dies geschieht mit Hilfe eines festen Testmusters, welches ein Bilderset und fixe Grenzwerte beinhaltet.

Vorbedingung:

- Die Evaluation wurde mit einem geeignetem Testset (verschiedene Auflösungen und jeweils verschiedene Bildqualitäten) durchgeführt.

Nachbedingung:

- Die Evaluation wurde durchgeführt und ausführlich inkl. verwendeter Tools und entsprechenden Vergleichsbedingungen im Endbericht dargestellt.
- Die Ergebnisse wurden in einer Präsentation vorgestellt.

TASKS	VERANTW.	ERL.	S.G.	S.V.
Überlegen eines Grobkonzeptes für die Evaluation	Marcel & Sören	X	n.A.	n.A.
Refactoring des Konzepts mit dem Betreuer	Marcel & Sören	X	n.A.	n.A.
Referanzplakate organisieren	Marcel & Sören	X	n.A.	n.A.
Referenzbilder aufbereiten	Marcel & Sören	X	n.A.	n.A.
Testsets zusammenstellen	Marcel & Sören	X	n.A.	n.A.
Algorithmentest implementieren	Marcel & Sören	X	n.A.	n.A.
Auswertung erstellen	Marcel & Sören	X	2	n.A.
Evaluation der Aufnahmeauflösung	-	-	n.A.	n.A.
Evaluationsauswertung entwickeln	Marcel & Dennis	X	11	12
Evaluation durchführen, Präsentation erarbeiten und vorstellen	Marcel & Dennis	X	8	4

Evaluation mit Menschen

Lfd-Nummer: 9	Priorität:	
Evaluation mit Menschen		
Beginn: 19.02.2008	Risiko: niedrig	Zeit: 20 h

Beschreibung

Die Entwickler haben nur eine eingeschränkte und starre Sicht auf das entwickelte Produkt und die Akzeptanz des Endkunden nach der Fertigstellung fehlt. Um eine hohe Akzeptanz durch den Benutzer zu gewährleisten, muss die Anwendung von real existierenden Personen evaluiert werden. Dadurch werden Defizite in der Bedienung aufgedeckt und Entwicklungsbedarf ermittelt. Hierzu wird ein Konzept entwickelt, das die zu befragende Zielgruppe, Fragebögen, sowie nötige Erweiterungen des Handy-Clients beeinhaltet.

Vorbedingung:

- Keine Vorbedingungen.

Nachbedingung:

- Die qualitativen Ergebnisse der Evaluation wurden im Endbericht festgehalten.
- Die Ergebnisse wurden in einer Präsentation vorgestellt.

Tasks	Verantw.	Erl.	S.g.	S.v.
Überlegen eines Konzeptes zur Evaluation	Marcel & Sören	X	n.A.	n.A.
Refactoring des Konzepts mit dem Betreuer	Marcel & Sören	X	n.A.	n.A.
Produktion eines Flyers, um die Ziele der PG für externe Personen darzustellen	Sören	X	n.A.	n.A.
Dummy-Bookmarks einfügen	Marcel & Sören & Torben	X	n.A.	n.A.
Menü-Engine, die Menüpunkte für Bookmarks dynamisch erzeugt	Torben & Christian M.	X	n.A.	n.A.
Fragebogen überarbeiten (5-Punkt-Skala)	Sören	X	n.A.	n.A.
Absprache mit Betreuern	Sören & Marcel	X	n.A.	n.A.
Aufbereitung und Präsentation der Evaluationsergebnisse	Sören & Chr. N. & Jeschua	X	20	25

Client GUI anpassen

Lfd-Nummer: 10	Priorität:	
Client GUI anpassen		
Beginn: 04.03.2008	Risiko: hoch	Zeit: 18 h

BESCHREIBUNG
Die Attraktivität einer Anwendung für einen Benutzer steigt durch eine intuitive Bedienung, sowie einer optisch ansprechende Gestaltung. Durch ständige Erweiterungen und Strukturumstellungen muss der Handy-Client fortlaufend angepasst werden. Die grafische Oberfläche des Handy-Clients wird konsequent einen einheitlichen Gestaltungsstil verwenden und ansprechend gestaltet sein. Alle Elemente des UIs werden nach demselben Stil aufgebaut, gestaltet und bedienbar sein. Der optische Aufbau wird sich an anerkannte Standards des Software Engineerings halten. So können alle Funktionalitäten innerhalb von maximal drei bis vier Schritten erreicht werden.

Vorbedingung:

- Keine Vorbedingungen.

Nachbedingung:

- Der Handy-Client besitzt einen einheitlichen Gestaltungsstil.
- Der Entwurf für das grafische Layout des Clients wurde präsentiert.
- Alle Anwendungsfunktionen sind in maximal drei bis vier Schritten erreichbar.
- Die Ergebnisse der Zoom-Implementierung wurden in einer Präsentation vorgestellt.
- Die ergänzte Hilfe zur 90°-Drehung wurde in einer Live-Demo vorgestellt.

TASKS	VERANTW.	ERL.	S.G.	S.V.
Ladebalken umgestalten (keine Nummern mehr anzeigen, „Sanduhr“)	Christian M. & Christian N.	X	n.A.	n.A.
Einen Entwurf zur grafischen Gestaltung des Handy-Clients erstellen	Jeschua	X	n.A.	n.A.
Die Elemente der Handy-GUI auf eine einheitliche Gestaltung überprüfen	Jeschua	X	n.A.	n.A.
Die Elemente der Handy-GUI an eine einheitlich Gestaltung anpassen			n.A.	n.A.

Lfd-Nummer: 10	Priorität:			
Client GUI anpassen (Forts.)				
TASKS	VERANTW.	ERL.	S.G.	S.V.
Skalierung der aufgenommenen Fotos auf die Größe des verwendeten Handy-Displays	Christian M. & Christian N.	X	n.A.	n.A.
GUI verschnellern, Speicherverbrauch minimieren	alle	X	n.A.	n.A.
Refactoring allgemein und speziell Camera-MIDlet	alle	X	n.A.	n.A.
J2ME-Polish auf neue Version updaten	Christian N. & Jeschua	X	n.A.	n.A.
Hilfe-Bug reparieren	Jeschua, Jan	X	n.A.	n.A.
IP-Eingabe überarbeiten	Chritian N. & Jeschua	X	5	5
Bug (Thread-Exception) in „Bild versenden“ beheben (tritt auf, wenn man zurück will)	Oliver & Jan	X	n.A.	n.A.
Zoom-Funktionalität	Oliver & Jan	X	15	22
Benutzerhilfe für 90° Drehung um bessere Bilder erhalten zu können	Oliver & Jan	-	3	-

Aufbau kontextueller Bookmarks

Lfd-Nummer: 13a	Priorität:	
Aufbau kontextueller Bookmarks		
Beginn: 04.03.2008	Risiko: hoch	Zeit: 6 h

BESCHREIBUNG
Um Hintergrundinformationen und weitere Dienste zu Plakaten zu erhalten, wird der Benutzer über das Versenden von gemachten Fotos kontextuelle Bookmarks erstellen und anfragen können, die ihm weiterführende und evtl. anschließend verfügbare Dienste zur Verfügung stellen. Die Problematik besteht darin, ein Konzept zur Gestaltung und Repräsentation von kontextuellen Bookmarks zu finden und ggf. zu erstellen. Hierzu wird zunächst nach bereits existierenden Lösungsansätzen recherchiert und darauf basierend ein Konzept entworfen, wie durch den Benutzer angefragte Bookmarks in sich aufgebaut sind und wie sie ihm präsentiert werden. Nach Erstellung des Konzeptes muss dieses in der Anwendung umgesetzt werden.

Vorbedingung:

- Keine Vorbedingungen.

Nachbedingung:

- Es wurde ein Konzept zum Aufbau kontextueller Bookmarks erstellt und in einer Präsentation vorgestellt.

TASKS	VERANTW.	ERL.	S.G.	S.V.
„State of the art“- Analyse bestehender Techniken	Christian M. & Christian N.	X	n.A.	n.A.
Konzeptentwurf zum Aufbau eines Bookmarks	Jan & Oliver & Christian M.	X	n.A.	n.A.
Konzeptentwurf zur Repräsentation von Bookmarks auf dem Handy-Client	Dennis & Jan & Christian N.	X	n.A.	n.A.
Präsentation „Kontextuelle Bookmarks“ erstellen	Christian M. & Oliver & Dennis	X	6	4

Implementierung kontextueller Bookmarks

Lfd-Nummer: 13b	Priorität:			
Implementierung kontextueller Bookmarks				
Beginn: 04.03.2008	Risiko: hoch	Zeit: n.a.		
BESCHREIBUNG				
In Storycard 13a wurde ein Konzept zur Darstellung und Umsetzung von kontextuellen Bookmarks entworfen. Ziel ist es nun, dieses Konzept in der bestehenden Anwendung erfolgreich umzusetzen.				

Vorbedingung:

- Es wurde ein Konzept zum Aufbau kontextueller Bookmarks erstellt und in einer Präsentation vorgestellt.

Nachbedingung:

- Die Umsetzung des Konzeptes wird mit Hilfe einer Präsentation und Demonstration mit Hilfe der Anwendung dargestellt.

TASKS	VERANTW.	ERL.	S.G.	S.V.
Implementierung einer Datenstruktur für Bookmarks	Christian M. & Jeschua & Oliver & Torben	X	n.A.	n.A.
Laden/Speichern von kontextuellen Bookmarks auf dem Handy	Sören & Oliver	X	n.A.	n.A.

Dienste in Bookmarks

Lfd-Nummer: 14a	Priorität:	
Dienste in Bookmarks		
Beginn: 04.03.2008	Risiko: hoch	Zeit: 6 h

BESCHREIBUNG
Dem Benutzer sollen neben einfachen Hintergrundinformationen zu fotografierten Plakaten auch weitere externe Dienste und Dienstleistungen angeboten werden. So müssen die vom Benutzer angefragten Bookmarks mit externen Diensten und Dienstleistungen verknüpft werden. Hierzu muss zunächst ein Lösungsansatz zur Einbettung von Diensten und Dienstleistungen entwickelt und testweise mit Mockup-Diensten (fest im Programmiercode festgelegte Dienste) implementiert werden.

Vorbedingung:

- Keine Vorbedingungen.

Nachbedingung:

- Es wurde ein Konzept zur Einbettung von Diensten und Dienstleistungen in kontextuelle Bookmarks entwickelt und in einer Präsentation vorgestellt.
- Die Implementierung von Mockup-Diensten wird mit Hilfe der Anwendung demonstriert.

TASKS	VERANTW.	ERL.	S.G.	S.V.
Konzept zur Einbettung von Diensten/-Dienstleistungen in Bookmarks erstellen	???	X	n.A.	n.A.
Realisierung des erstellten Konzeptes mit Mockup-Diensten	???	X	n.A.	n.A.
Konzeptentwurf Datenbank für Dienste	Christian M. & Dennis	X	n.A.	n.A.
Polish-Browser für Dienst-URLs ausprobieren	Jan & Sören & Oliver	X	n.A.	n.A.
Präsentation „Dienste in Bookmarks“	Christian M. & Oliver & Dennis	X	6	6

Dienste in Bookmarks implementieren

<table>
<tr><td colspan="2">Lfd-Nummer: 14b</td><td colspan="3">Priorität:</td></tr>
<tr><td colspan="5">Dienste in Bookmarks implementieren</td></tr>
<tr><td>Beginn: 04.03.2008</td><td>Risiko: hoch</td><td colspan="3">Zeit: 11 h</td></tr>
<tr><td colspan="5">BESCHREIBUNG</td></tr>
<tr><td colspan="5">In SC 10.3.1 wurden Konzepte zur Realisierung von Diensten in Bookmarks erstellt und mit fest einprogrammierten Diensten implementiert. Nun soll diese Implementierung durch die Nutzung von in eine Datenbank eingepflegte Dienste (real existierende Dienste) erweitert werden.</td></tr>
<tr><td colspan="5">Vorbedingung:
• Es wurde ein Konzept zur Einbettung von Diensten und Dienstleistungen in kontextuelle Bookmarks entwickelt und in einer Präsentation vorgestellt.
Nachbedingung:
• Die nun in der Datenbank vorhandenen Dienste werden durch den Client in einer Live-Demo abgefragt und ihre Realisierung in einer Präsentation vorgestellt.</td></tr>
<tr><td>TASKS</td><td>VERANTW.</td><td>ERL.</td><td>S.G.</td><td>S.V.</td></tr>
<tr><td>Auf real existierende Dienste anpassen</td><td>Christan M. & Torben</td><td>X</td><td>11</td><td>12</td></tr>
<tr><td>Erstellung einer Präsentation zu real existierenden Diensten</td><td>Christan M. & Torben</td><td>-</td><td>-</td><td>-</td></tr>
</table>

Personalisierung

Lfd-Nummer: 17a	Priorität:	
Personalisierung		
Beginn: 31.3.2008	Risiko: hoch	Zeit: 52 h

BESCHREIBUNG
Die dem Benutzer angebotenen Bookmarks sollen personalisiert werden, um die Akzeptanz für die angebotenen Dienste zu erhöhen. Verschiedene Verfahren zur Personalisierung stehen zur Verfügung und bieten Vor- und Nachteile. Es muss evaluiert werden, welche Personalisierungsverfahren in diesem Fall am besten geeignet sind und wie die Personalisierung umgesetzt werden kann.

Vorbedingung:

- keine Vorbedingungen.

Nachbedingung:

- Eine „state-of-the-art“ Analyse wurde durchgeführt, ein Verfahren wurde darauf basierend ausgewählt und in einer Präsentation vorgestellt.

TASKS	VERANTW.	ERL.	S.G.	S.V.
Recherche möglicher Personalisierungsverfahren und Präsentation erstellen	Christian M. & Christian N. & Oliver	X	10	12
Entscheiden, welche Verfahren geeignet sind	alle	X	9	9
Konzept zur Identifikation des Clients beim Server	Dennis	X	4	4
Konzept: Historie bisher aufgerufener Dienste für jeden Benutzer	Christian M.	X	10	2
Konzept zur Personalisierung erarbeiten	Torben & Christian M.	X	19	18

Personalisierung implementieren

Lfd-Nummer: 17b	Priorität:	
Personalisierung implementieren		
Beginn: 31.3.2008	Risiko: hoch	Zeit: 118 h

Beschreibung
In SC 10.3.1 wurde ein Konzept zur Personaliserung von Bookmarks erarbeitet. Dieses wird nun anhand der dort getroffenen Designentscheidungen implementiert.

Vorbedingung:

- Eine „state-of-the-art“ Analyse wurde durchgeführt, ein Verfahren wurde darauf basierend ausgewählt und in einer Präsentation vorgestellt.

Nachbedingung:

- Die Implementierung der Personalisierung wurde anhand einer Live-Demo und einer Präsentation vorgestellt.

Tasks	Verantw.	Erl.	S.G.	S.V.
Implementierung der Personalisierung	Christian M. & Dennis & Torben		118	30
Erstellung einer Präsentation zur Personalisierung	Christian M. & Dennis & Torben	X	-	3

Unbekannte Objekte annotieren

Lfd-Nummer: 20	Priorität:	
Unbekannte Objekte annotieren		
Beginn: 12.05.2008	Risiko:	Zeit: 20 h

BESCHREIBUNG
Bisher erhält der Benutzer als Rückmeldung, wenn ein gesendetes Foto zu keinem Plakat zugeordnet werden konnte, lediglich eine diesbezügliche Hinweismeldung. Ziel dieser Storycard ist es, dem Benutzer die Möglichkeit zu geben, Fotos zu annotieren. Hierzu soll er Kampagnen zum Foto auswählen, Dienste zuordnen und das Foto mit Metadaten anreichern können. Diese Daten kann er anschließend persistent in der Datenbank und für andere Nutzer zugänglich speichern.

Vorbedingung:

- Datenstruktur zur Speicherung von Plakaten und Bookmarks muss implementiert worden sein.

Nachbedingung:

- Ein Konzept zur Speicherung und Annotation von zuvor unbekannten Objekten wurde erstellt und in einer Präsentation vorgestellt.
- Das Konzept wurde implementiert und in einer Live-Demo vorgestellt.

TASKS	VERANTW.	ERL.	S.G.	S.V.
Konzepterstellung für die Annotation unbekannter Objekte und Präsentation	Oliver & Jan	X	5	7
Implementierung des Konzeptes zur Annotation unbekannter Objekte	Oliver & Jan	-	15	-

10.3.2. Umsetzung

Betrachtete Algorithmen

Es sollte zunächst eine einfache, regelbasierte Personalisierung durchgeführt werden, die dann um eine inhaltsbasierte Personalisierung mittels Lucene ergänzt werden sollte.

Für die regelbasierte Personalisierung wurde zunächst die Historie der Dienstaufrufe durch jeden Benutzer genutzt, die der Server führt. Dabei war jeder Dienst einer Kategorie zugeteilt. So konnten – nach einer gewissen Anzahl and Dienstaufrufen – diese Daten benutzt werden, um eine Statistik der beliebtesten Dienstarten für jeden Benutzer zu erstellen. Auf diese Weise konnte die Auslieferung neuer Bookmarks an das zu erwartende Verhalten des Nutzers angeglichen werden. In einer einfachen Fassung des

Systems ließen sich so Dienste in der Liste nach oben sortieren, deren Kategorie bei dem Benutzer beliebt ist.

Weitere betrachtete Möglichkeiten einer Filterung bestanden anhand dieser Statistik, so dass Dienste besonders unbeliebter Kategorien gar nicht erst an den Benutzer ausgeliefert werden würden. Alternativ könnte man diese auch mit einem Attribut als „versteckt“ markieren und dann im Client ausblenden, der Benutzer hätte aber die Möglichkeit, sie durch Auswahl eines Menüpunktes wieder anzeigen zu lassen.

Anpassung der XML-Struktur

Für die Übertragung der vollständigen Bookmarks wurde eine Struktur gewählt, bei der an einem Wurzelelement „olivanda“, das über Attribute wie eine eindeutige ID und einen Titel (des Kontextuellen Bookmarks) verfügte, ein „services“-Element mit beliebig vielen „service“-Subelementen hing. Diese wiederum waren mit den für Dienste typischen Attributen, also Dienst-ID, Beschreibungstext und URL ausgestattet.

Diese Struktur bot zunächst eine gute Flexibilität bezüglich der zu übertragenden Dienste, was für die präzise Modellierung der Bookmarks nötig war. Leider konnte das in Release II, Iteration III entwickelte XML-Template nicht mehr genutzt werden, da bei seiner Nutzung keine dynamische Erzeugung der „service“-Elemente möglich war.

Erweiterung des Datenbankschemas

Zur Pflege der Plakate und Bookmarks durch potentielle Kunden musste eine Möglichkeit geschaffen werden, diesen Anbietern ein Login in das Wartungsfrontend zu ermöglichen. Dabei sollte aber auch ein Rechtesystem umgesetzt werden, damit keine Änderungen an Werbekampagnen durch Dritte möglich sein würden. Dazu wurde eine eigene Tabelle mit Anbieter-Logins erstellt und die entsprechenden Daten mittels Anbieter-IDs konkreten Anbietern zugeordnet.

Bei den Diensten wurde eine zusätzliche 1:n-Relation zu einer Tabelle mit Metadaten geschaffen, um flexibel Metadaten mit den Diensten verknüpfen zu können. Dies ermöglichte eine sinnvolle Umsetzung der inhaltsbasierten Personalisierung mittels Lucene.

Außerdem wurden bei einigen Tabellen zusätzliche Attribute eingefügt, beispielsweise eine Laufzeit bei Kampagnen. Die Erweiterung der Benutzertabelle um Schlüsselworte für die Lucene-Suche stand zum Ende der Iteration noch aus. Einige der erwähnten Änderungen verursachten ihrerseits eine Inkompatibilität der Datenbankschnittstelle des Olivanda-Servers mit dem Datenbankschema, so dass eine Anpassung der Datenbankschnittstelle nötig wurde, um den Server wieder funktionstüchtig zu machen.

Die im Zuge der Entwicklung des Frontends revidierte Datenstruktur ist in Abblidung 10.2 dargestellt.

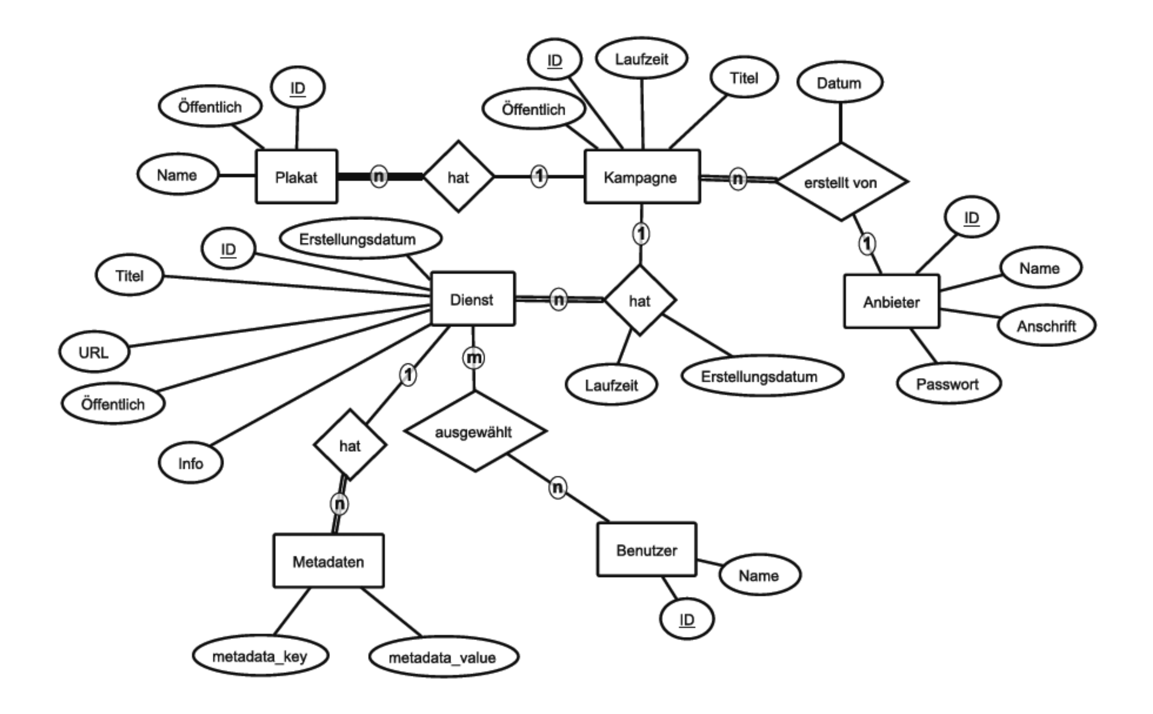


Abbildung 10.2.: ER-Diagramm der Datenbankstruktur nach der Revision

Frontend für kontextuelle Bookmarks

Wie bereits in Abschnitt 10.3.2 erwähnt, bestand die Notwendigkeit einer Benutzer- bzw. Anbieterschnittstelle der Datenbank. Auf diese Weise sollte es Anbietern beziehungsweise Kunden ermöglicht werden, eigene Kampagnen samt Diensten und Plakaten in die Software einpflegen zu können.

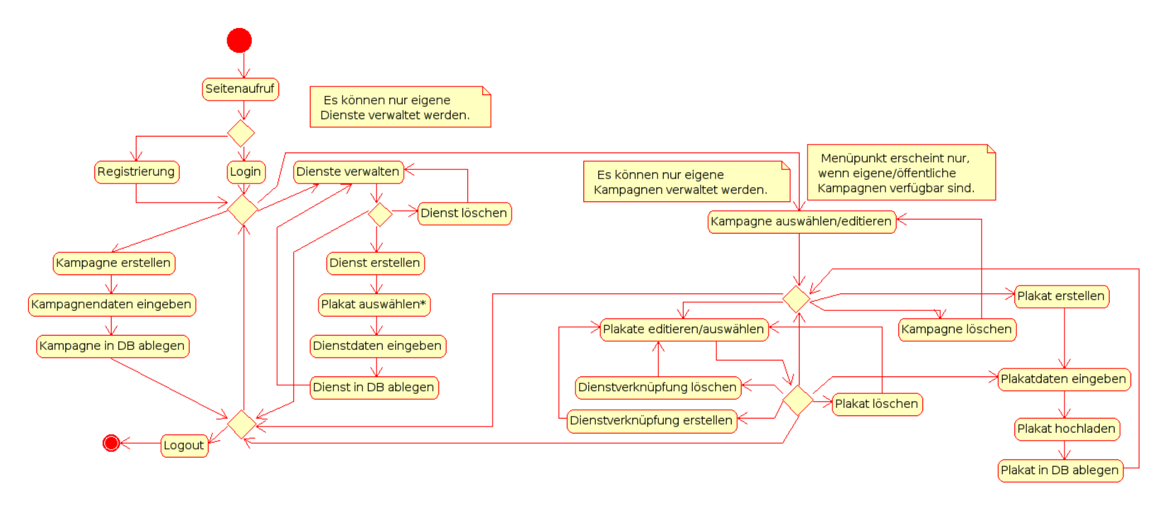
Die Kunden sollten in der Lage sein, sich im Webinterface zu registrieren und anzumelden, um einerseits eine eindeutige Zuordnung bezüglich Änderungen zu ermöglichen und andererseits um das gewünschte Rechtesystem zu implementieren. Hierfür wurde eine ergonomische und optisch ansprechende GUI benötigt, welche es dem Benutzer ermöglichte, sämtliche Vorgänge möglichst intuitiv und ohne Anleitung durchzuführen. Die Kunden sollten in der Lage sein, Kampagnen, Plakate und Dienste hinzuzufügen und sie zu editieren sowie zu löschen, sofern sie vorher von ihnen angelegt wurden.

Veranschaulicht wird der gewünschte Vorgang in Abbildung 10.3.2.

Für die Realisierung standen eine Webplattform und eine Client/Server-Architektur zur Auswahl. Die globale Erreichbarkeit sowie die Tatsache, dass keine Installation und Updates beim Kunden nötig sind, sprachen letzten Endes für die Online-Lösung. Weiterhin wurde diskutiert, in welcher Sprache das Frontend geschrieben werden sollte. Die Alternative PHP setzte sich gegenüber Java Server Pages und Ruby on Rails durch, da hier

die Resonanz der Projektgruppenmitglieder einerseits wegen vorhandener Vorkenntnisse, andererseits aufgrund des Wunsches, sich in PHP weiterzubilden, am stärksten war.

Um eine Trennung zwischen Logik und GUI einfach zu gestalten, wurde die PHP Template Engine „Smarty" eingesetzt. Sie bietet den Vorteil, dass gleichzeitig an den Bereichen der Logik und GUI gearbeitet werden kann. Weiterhin kann dieselbe GUI für verschiedene Logiken verwendet werden. Zusätzlich können durch Arbeiten an dem Design keine Fehler oder Sicherheitslücken im Programmcode entstehen. Die Funktionsweise von Smarty ist derart, dass die Logik in den eigentlichen PHP-Dateien geschrieben wird, die grafische Ausgabe erfolgt dann über Templatedateien (.tpl). Smarty bietet eine rudimentäre Logik, die lediglich zu Anzeigezwecken dient. Die benötigten Variablen zur Anzeige werden einem Smarty-Objekt zugewiesen und sind dann in den .tpl Dateien verfügbar.



Frontend zur Annotation unbekannter Objekte

Wird ein aufgenommenes Foto von unserem System nicht erkannt, soll es dem Anwender möglich sein, wie in Storycard 10.3.1 beschrieben eigene Fotos zu neuen bzw. bereits bestehenden Kampagnen zu annotieren. Wird das Foto nun im weiteren Verlauf erneut aufgenommen, wird es mit der zugehörigen Kampagne erkannt.

Um eine möglichst effiziente Programmierung ohne Codeduplizierung und einen geringen Integrationsaufwand in den Client zu ermöglichen, wurde hierfür auf eine webbasierte Lösung gesetzt, die ebenfalls die bereits unter 10.3.2 beschriebenen Logikfunktionalitäten nutzen konnte. Der Pageflow des Frontends ist unter 10.3.2 dargestellt.

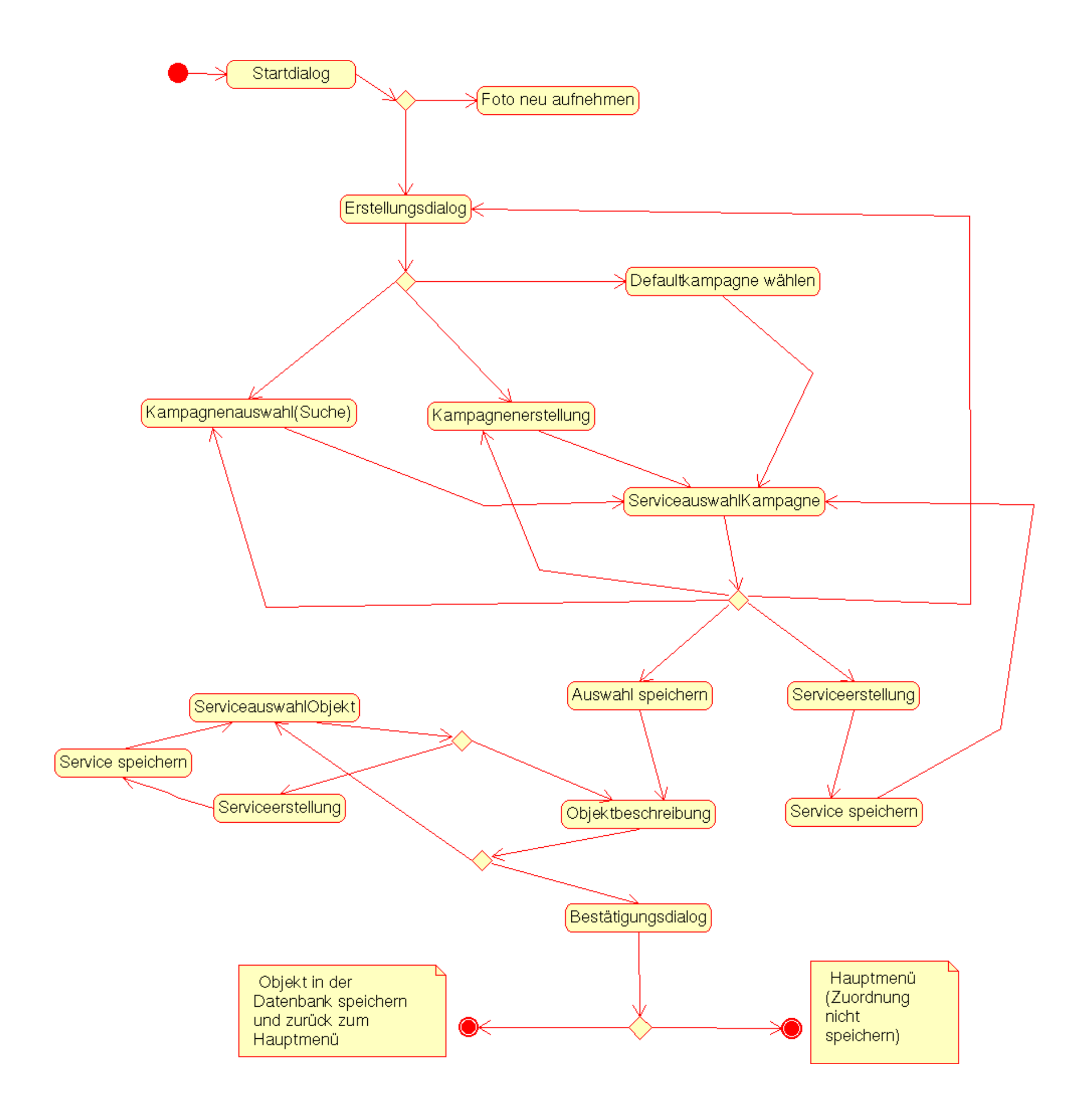


Abbildung 10.3.: Pageflow zur Annotation unbekannter Objekte

Systemarchitektur

Da in dieser Iteration vor allem neue Softwarekomponenten wie das Webfrontend bearbeitet wurden, ergaben sich an der Architektur der bestehenden Komponenten Olivanda-Server und -Client keine Änderungen.

10.3.3. Ergebnisse

Ergebnisse der Akzeptanztests

Die für diese Iteration geplanten Akzeptanztests wurden im ersten Anlauf nur teilweise erfolgreich durchgeführt, da einige Umsetzungen noch nicht abgeschlossen worden waren. Die Storycards 9 und 18a wurden erfolgreich beendet und ihr Akzeptanztest erfüllt. Die Annotation von Objekten (Storycard 20) und deren Speicherung innerhalb des Systems konnte exemplarisch an der Anhand der Aufnahme eines Stuhls demonstriert werden.

Retrospektive

Besonders zu erwähnen war die allgemeine Freude darüber, dass in dieser Iteration nach der langen Evaluationsphase wieder mit der Weiterentwicklung der Software begonnen wurde. Da von vielen Mitgliedern der Projektgruppe Programmierarbeiten gegenüber Dokumentations- und anderen Aufgaben bevorzugt wurden, wurde dies sehr positiv aufgenommen.

Sorge bestand allerdings bezüglich der zur Umsetzung des Webfrontends verfügbaren Zeit, da Konzeption und Umsetzung innerhalb einer Iteration durchgeführt werden sollten. Nachdem in der dritten Iterationswoche seitens der Betreuer noch Änderungen an der Datenstruktur und Verarbeitung angeregt wurden, war an eine Fertigstellung des Frontends nicht mehr zu denken.

Diese Verbesserungswünsche wurden insgesamt als Eingriff in den Iterationszyklus empfunden. Insbesondere, da sie erhebliche Änderungen in den bereits fertiggestellten bzw. in Umsetzung befindlichen Komponenten nötig machen würden. Zu guter Letzt ist anzufügen, dass sie die Dokumentationswoche stark beeinträchtigten, da bspw. die Arbeiten am Endbericht mit deutlich weniger Arbeitskraft durchgeführt werden mussten.

Zum ersten Mal wurde auch eine Überprüfung der Aufwandsabschätzungen für die erledigten Tasks durchgeführt. Dabei zeigte sich, dass für diese Tasks in Summe Zwölf Stunden zu viel eingeplant worden waren bzw. im Mittel 0,7 Stunden zu viel. Eine Tendenz, welche Tasks häufig von Fehlschätzungen betroffen waren, ließ sich nicht ermitteln. Auffällig war aber, dass bei sehr vielen Tasks der Aufwand geringfügig zu hoch eingeschätzt wurde, während bei einigen wenigen Tasks deutlich zu niedriger Aufwand angesetzt worden war.

10.4. Iteration II

Die zweite Iteration des dritten Releases fand in den letzten Wochen der Vorlesungszeit des Sommersemesters statt. Hauptaufgabe war die Durchführung der zweiten Evaluation, welche im Zusammenhang mit der Plakatkampagne von Ströer-Media stand.

Parallel dazu wurde die Umsetzung der kontextuellen Bookmarks und deren Parametrisierung vorangetrieben sowie das in Iteration I begonnene Webfrontend zur Pflege der Kampagnendaten weiter bearbeitet.

10.4.1. Storycards

Mit Ausnahme der neu eingeführten Storycard für die Ströer Medien-Evaluation bestanden die Storycards dieser Iteration aus Fortsetzungen existierender Storycards, die in der letzten Iteration nicht abgeschlossen werden konnten.

Ströer Medien - Evaluation

Lfd-Nummer: 21	Priorität: 1	
Ströer Medien - Evaluation		
Beginn: 19.06.2008	Risiko: mittel	Zeit: 153 h

Beschreibung
Im Kontext einer Kooperation mit der Firma Ströer Medien im Rahmen einer weiteren Veranstaltung wird Anfang Juli 2008 eine Plakatkampagne gestartet. Diese bietet die Möglichkeit, unsere Anwendung mit Hilfe von universitätsexternen Personen zu evaluieren. War bei der Evaluation 10.3.1 der Vergleich zwischen herkömmlichem Internetbrowser und dem „Point and Shoot "-Verfahren von Interesse, soll in diesem Rahmen nach weiteren Evaluationsmöglichkeiten gesucht werden, ein Konzept erstellt und dieses durchgeführt werden.

Vorbedingung:

- Keine Vorbedingungen.

Nachbedingung:

- Ein Konzept zur Evaluation wurde erstellt und in einer Präsentation vorgestellt.
- Die Evaluation wurde durchgeführt und die Ergebnisse in einer Präsentation vorgestellt.

Tasks	Verantw.	Erl.	S.G.	S.V.
Überprüfung der Handyfunktionalität auf UMTS	Jeschua & Sören	-	6	-
Erstellung eines Konzeptes zur Evaluation	Christian N. & Sören	x	37	36
Ausarbeitung eines Fragebogens	Christian N. & Sören & Marcel	-	20	33
Durchführung der Evaluation	Christian N. & Sören & Marcel & Jeschua	-	60	42
Auswertung der Evaluation und Erstellung einer Präsentation	Christian N. & Sören & Marcel & Jan	-	30	-

Client GUI anpassen

Lfd-Nummer: 10	Priorität: 1	
Client GUI anpassen		
Beginn: 04.03.2008	Risiko: niedrig	Zeit: 4 h

Beschreibung
Die Attraktivität einer Anwendung für einen Benutzer steigt durch eine intuitive Bedienung, sowie einer optisch ansprechende Gestaltung. Durch ständige Erweiterungen und Strukturumstellungen muss der Handy-Client fortlaufend angepasst werden. Die grafische Oberfläche des Handy-Clients wird konsequent einen einheitlichen Gestaltungsstil verwenden und ansprechend gestaltet sein. Alle Elemente des UIs werden nach demselben Stil aufgebaut, gestaltet und bedienbar sein. Der optische Aufbau wird sich an anerkannte Standards des Software Engineerings halten. So können alle Funktionalitäten innerhalb von maximal drei bis vier Schritten erreicht werden.

Vorbedingung:

- Keine Vorbedingungen.

Nachbedingung:

- Der Handy-Client besitzt einen einheitlichen Gestaltungsstil.
- Alle Anwendungsfunktionen sind in maximal drei bis vier Schritten erreichbar.
- Die überarbeiteten Hilfe wurde in einer Live-Demo vorgestellt.

Tasks	Verantw.	Erl.	S.g.	S.v.
Ladebalken umgestalten (keine Nummern mehr anzeigen, „Sanduhr“)	Christian M. & Christian N.	X	n.A.	n.A.
Einen Entwurf zur grafischen Gestaltung des Handy-Clients erstellen	Jeschua	X	n.A.	n.A.
Die Elemente der Handy-GUI auf eine einheitliche Gestaltung überprüfen	Jeschua	X	n.A.	n.A.
Die Elemente der Handy-GUI an eine einheitlich Gestaltung anpassen			n.A.	n.A.
Skalierung der aufgenommenen Fotos auf die Größe des verwendeten Handy-Displays	Christian M. & Christian N.	X	n.A.	n.A.
GUI verschnellern, Speicherverbrauch minimieren	alle	X	n.A.	n.A.

Lfd-Nummer: 10	Priorität:			
Client GUI anpassen (Forts.)				
TASKS	VERANTW.	ERL.	S.G.	S.V.
Refactoring allgemein und speziell Camera-MIDlet	alle	X	n.A.	n.A.
J2ME-Polish auf neue Version updaten	Christian N. & Jeschua	X	n.A.	n.A.
Hilfe-Bug reparieren	Jeschua, Jan	X	n.A.	n.A.
IP-Eingabe überarbeiten	Chritian N. & Jeschua	X	5	5
Bug (Thread-Exception) in „Bild versenden“ beheben (tritt auf, wenn man zurück will)	Oliver & Jan	X	n.A.	n.A.
Zoom-Funktionalität	Oliver & Jan	X	15	22
Die Handy-Hilfemenüs auf den aktuellen Stand bringen (u.a. 90° Drehung des Handys)	Oliver & Jan	-	4	-

Implementierung kontextueller Bookmarks

Lfd-Nummer: 13b	Priorität: 2	
Implementierung kontextueller Bookmarks		
Beginn: 04.03.2008	Risiko: hoch	Zeit: 51 h

BESCHREIBUNG
In Storycard 13a wurde ein Konzept zur Darstellung und Umsetzung von kontextuellen Bookmarks entworfen. Ziel ist es nun, dieses Konzept in der bestehenden Anwendung erfolgreich umzusetzen.

Vorbedingung:

- Es wurde ein Konzept zum Aufbau kontextueller Bookmarks erstellt und in einer Präsentation vorgestellt.

Nachbedingung:

- Die Umsetzung des Konzeptes wird mit Hilfe einer Präsentation und Demonstration mit Hilfe der Anwendung dargestellt.
- Die Demonstration läuft folgendermaßen ab: Der Benutzer nimmt ein Foto auf und sendet dieses an den Server. Dieser ermittelt den Standort des Benutzers und gibt kontextbezogene Bookmarks zu dem erkannten Foto zurück. Dieser Vorgang ist von Benutzern mit unterschiedlichem Profil zu wiederholen, damit der kontextuelle Hintergrund gezeigt werden kann.

TASKS	VERANTW.	ERL.	S.G.	S.V.
Implementierung einer Datenstruktur für Bookmarks	Christian M. & Jeschua & Oliver & Torben	X	n.A.	n.A.
Laden/Speichern von kontextuellen Bookmarks auf dem Handy	Sören & Oliver	X	n.A.	n.A.
Zuvor angepasstes Konzept von kontextuellen Bookmarks implementieren	Dennis & Torben & Jan & Oliver	-	51	63

Konzept zur Personalisierung

Lfd-Nummer: 17a	Priorität: 2	
Personalisierung		
Beginn: 31.3.2008	Risiko: niedrig	Zeit: 6 h

Beschreibung
Die dem Benutzer angebotenen Bookmarks sollen personalisiert werden, um die Akzeptanz für die angebotenen Dienste zu erhöhen. Verschiedene Verfahren zur Personalisierung stehen zur Verfügung und bieten Vor- und Nachteile. Es muss evaluiert werden, welche Personalisierungsverfahren in diesem Fall am besten geeignet sind und wie die Personalisierung umgesetzt werden kann.

Vorbedingung:

- Eine „state-of-the-art“ Analyse wurde durchgeführt, ein Verfahren wurde darauf basierend ausgewählt und in einer Präsentation vorgestellt.
- Der Aufbau von kontextuellen Bookmarks wurde überarbeitet.

Nachbedingung:

- Das vorhandene Konzept zur Personalisierung wurde auf das überarbeitete Konzept zu kontextuellen Bookmarks hin angepasst und in einer Präsentation vorgestellt

Tasks	Verantw.	Erl.	S.G.	S.V.
Recherche möglicher Personalisierungsverfahren und Präsentation erstellen	Christian M. & Christian N. & Oliver	X	10	12
Entscheiden, welche Verfahren geeignet sind	alle	X	9	9
Konzept zur Identifikation des Clients beim Server	Dennis	X	4	4
Konzept: Historie bisher aufgerufener Dienste für jeden Benutzer	Christian M.	X	10	2
Konzept zur Personalisierung erarbeiten	Torben & Christian M.	X	19	18
Das Konzept zur Personalisierung auf Basis der angepassten kontextuellen Bookmarks überarbeiten und Präsentation erstellen	Christian M. & Oliver	X	6	6

Personalisierung implementieren

Lfd-Nummer: 17b	Priorität: 2	
Personalisierung implementieren		
Beginn: 31.3.2008	Risiko: hoch	Zeit: 88 h

BESCHREIBUNG
In SC 10.3.1 wurde ein Konzept zur Personaliserung von Bookmarks erarbeitet. Dieses wird nun anhand der dort getroffenen Designentscheidungen implementiert.

Vorbedingung:

- Eine „state-of-the-art“ Analyse wurde durchgeführt, ein Verfahren wurde darauf basierend ausgewählt und in einer Präsentation vorgestellt.

Nachbedingung:

- Die Implementierung der Personalisierung wurde anhand einer Live-Demo und in einer Präsentation vorgestellt.
- Die Live-Demo ist äquivalent zu dem Ablauf des Akzeptanztests der SC 10.4.1.

TASKS	VERANTW.	ERL.	S.G.	S.V.
Implementierung der Personalisierung	Christian M. & Dennis & Torben		118	30
Erstellung einer Präsentation zur Personalisierung	Christian M. & Dennis & Torben	X	-	3

Frontend für kontextuelle Bookmarks

Lfd-Nummer: 19	Priorität: 3
Frontend für kontextuelle Bookmarks	

Beginn: 12.05.2008	Risiko: niedrig	Zeit: 68 h

BESCHREIBUNG

Dem Kunden der Anwendung ist es möglich, seine Plakatdaten in das bestehende System einzupflegen. Daher wird eine grafisch ansprechende und zu allen Funktionen in drei bis vier Schritten erreichbare Anwendung entworfen und implementiert werden. Der Kunde kann reine Bilddaten, dazu passende Services und Metadaten für seine Plakatkampagne über das Frontend einstellen. Diese sind dann automatisch für jeden Client verfügbar.

Vorbedingung:

- Die endgültige Datenstruktur für kontextuelle Bookmarks muss festgelegt sein
- Frontend muss Zugriff auf die Datenbank des Olivanda-Projektes besitzen

Nachbedingung:

- Ein Plakatdatensatz kann in das aktiv laufende System mit Hilfe des Frontends eingebunden werden. Hierzu zählen Bilddaten, kontextuelle Bookmarks sowie genutzte Dienste des Plakaterstellers.
- Die Ergebnisse werden in einer Präsentation und in einer Live-Demo vorgestellt. Die durch das Frontend beispielhaft eingestellte Plakatkampagne wird über den Handy-Client durch Aufnahme des entsprechenden Plakates abgerufen und mit den entsprechenden Bookmarks dargestellt.

TASKS	VERANTW.	ERL.	S.G.	S.V.
Recherche nach Realisierungsmöglichkeiten und bestehenden Technologien	-	X	7	4
Technologieauswahl	-	X	9	4
Konzept einer Datenstruktur	-	-	16	11
Konzept der Benutzeroberfläche	-	-	10	19
Implementierung der Datenstruktur	-	-	34	5
Implementierung des Frontends	-	-	26	27
Methoden der Datenbankschnittstelle ausfüllen	-	-	4	-
GUI vervollständigen und ausbauen	-	-	30	-
Besitzer zu Objekten hinzufügen	-	-	5	-

Unbekannte Objekte annotieren

Lfd-Nummer: 20	Priorität: 4			
Unbekannte Objekte annotieren				
Beginn: 12.05.2008	Risiko: mittel	Zeit: 29 h		
BESCHREIBUNG				
Bisher erhält der Benutzer als Rückmeldung, wenn ein gesendetes Foto zu keinem Plakat zugeordnet werden konnte, lediglich eine diesbezügliche Hinweismeldung. Ziel dieser Storycard ist es, dem Benutzer die Möglichkeit zu geben, Fotos zu annotieren. Hierzu soll er Kampagnen zum Foto auswählen, Dienste zuordnen und das Foto mit Metadaten anreichern können. Diese Daten kann er anschließend persistent in der Datenbank und für andere Nutzer zugänglich speichern.				

Vorbedingung:

- Datenstruktur zur Speicherung von Plakaten und Bookmarks muss implementiert worden sein.
- Ein Konzept zur Speicherung und Annotation von zuvor unbekannten Objekten wurde erstellt und in einer Präsentation vorgestellt.

Nachbedingung:

- Das Konzept wurde implementiert und in einer Live-Demo vorgestellt.

TASKS	VERANTW.	ERL.	S.G.	S.V.
Konzepterstellung für die Annotation unbekannter Objekte und Präsentation	Oliver & Jan	X	5	7
Implementierung des Konzeptes zur Annotation unbekannter Objekte	Oliver & Jan	-	44	15

Evaluation der Algorithmen

Lfd-Nummer: 8	Priorität: 4	
Evaluation der Algorithmen		
Beginn: 19.02.2008	Risiko: niedrig	Zeit: 37 h

Beschreibung

Die Algorithmen werden anhand von verschiedenen Gesichtspunkten, wie Geschwindigkeit, Treffergenauigkeit und Robustheit einheitlich getestet und bewertet. Dies geschieht mit Hilfe eines festen Testmusters welches ein Bilderset und fixe Grenzwerte beinhaltet.

Vorbedingung:

- Die Evaluation wurde mit einem geeignetem Testset (verschiedene Auflösungen und jeweils verschiedene Bildqualitäten) durchgeführt.

Nachbedingung:

- Die Evaluation wurde durchgeführt und ausführlich inkl. verwendeter Tools und entsprechenden Vergleichsbedingungen im Endbericht dargestellt.
- Die Funktionsfähigkeit der Algorithmenevaluation wird anhand der Evaluation eines exemplarischen Testsets demonstriert.

Tasks	Verantw.	Erl.	S.g.	S.v.
Überlegen eines Grobkonzeptes für die Evaluation	Marcel & Sören	X	n.A.	n.A.
Refactoring des Konzepts mit dem Betreuer	Marcel & Sören	X	n.A.	n.A.
Referanzplakate organisieren	Marcel & Sören	X	n.A.	n.A.
Referenzbilder aufbereiten	Marcel & Sören	X	n.A.	n.A.
Testsets zusammenstellen	Marcel & Sören	X	n.A.	n.A.
Algorithmentest implementieren	Marcel & Sören	X	n.A.	n.A.
Auswertung erstellen	Marcel & Sören	X	2	n.A.
Evaluation der Aufnahmeauflösung	-	-	n.A.	n.A.
Evaluationsauswertung entwickeln	Marcel & Dennis	X	11	12
Evaluation durchführen, Präsentation erarbeiten und vorstellen	Marcel & Dennis	X	8	4
Testsets erstellen (verschiedene Auflösungen, ähnliche Plakate)	-	-	20	-
Refactoring der Evaluationsimplementierung	Dennis	-	20	18
Evaluation erneut durchführen und Präsentation erstellen	-	-	11	-

Textdetektion und Texterkennung integrieren

Lfd-Nummer: 18b	Priorität: 4	
Textdetektion und Texterkennung implementieren		
Beginn: 12.05.2008	Risiko: hoch	Zeit: 54 h
Beschreibung		
In SC 10.3.1 wurden verschiedene Alternativen und Ergänzungen zu der bereits im Projekt realisierten SIFT-Variante diskutiert. Als weitere Vorgehensweise wurde beschlossen, Textdetektion und Texterkennung in das Projekt zu integrieren.		

Vorbedingung:

- Alternative Techniken wurden recherchiert und diskutiert

Nachbedingung:

- Es wurde ein Konzept zur Kombination von SIFT mit Textdetektion und Texterkennung in einer Präsentation vorgestellt.
- Texterkennung und Detektion wurden implementiert und in einer Live-Demo vorgestellt, diese umfasst alle im Konzept definierten Funktionalitäten.

Tasks	Verantw.	Erl.	S.G.	S.V.
Auswahl des zu nutzenden Frameworks für die Textdetektion und Texterkennung	Torben & Jan	-	4	-
Erstellung eines Konzeptes zur Integration von Textdetektion und Texterkennung in die bereits bestehende SIFT-Architektur erarbeiten und eine Präsentation erstellen	-	-	-	-
Implementierung von Textdetektion und Texterkennung im Projekt	Torben & Jan	-	50	-

Aufbau kontextueller Bookmarks

Lfd-Nummer: 13a	Priorität: 0 (abgeschlossen)	
Aufbau kontextueller Bookmarks		
Beginn: 04.03.2008	Risiko: niedrig	Zeit: 25 h

Beschreibung

Um Hintergrundinformationen und weitere Dienste zu Plakaten zu erhalten, wird der Benutzer über das Versenden von gemachten Fotos kontextuelle Bookmarks erstellen und anfragen können, die ihm weiterführende und evtl. anschließend verfügbare Dienste zur Verfügung stellen. Die Problematik besteht darin, ein Konzept zur Gestaltung und Repräsentation von kontextuellen Bookmarks zu finden und ggf. zu erstellen. Hierzu wird zunächst nach bereits existierenden Lösungsansätzen recherchiert und darauf basierend ein Konzept entworfen, wie durch den Benutzer angefragte Bookmarks in sich aufgebaut sind und wie sie ihm präsentiert werden. Nach Erstellung des Konzeptes muss dieses in der Anwendung umgesetzt werden.

Vorbedingung:

- Keine Vorbedingungen.

Nachbedingung:

- Es wurde ein überarbeitetes Konzept zum Aufbau kontextueller Bookmarks erstellt und in einer Präsentation vorgestellt.

Tasks	Verantw.	Erl.	S.G.	S.V.
„State of the art“- Analyse bestehender Techniken	Christian M. & Christian N.	X	n.A.	n.A.
Konzeptentwurf zum Aufbau eines Bookmarks	Jan & Oliver & Christian M.	X	n.A.	n.A.
Konzeptentwurf zur Repräsentation von Bookmarks auf dem Handy-Client	Dennis & Jan & Christian N.	X	n.A.	n.A.
Präsentation „Kontextuelle Bookmarks“ erstellen	Christian M. & Oliver & Dennis	X	6	4
Angepasstes Konzept für kontextuelle Bookmarks auf Basis der Kundenwünsche erstellen	Torben & Jeschua & Jan	X	25	25

Dienste in Bookmarks implementieren

Lfd-Nummer: 14b	Priorität:	
Dienste in Bookmarks implementieren		
Beginn: 04.03.2008	Risiko: hoch	Zeit: 11 h

Beschreibung
In SC 10.3.1 wurden Konzepte zur Realisierung von Diensten in Bookmarks erstellt und mit fest einprogrammierten Diensten implementiert. Nun soll diese Implementierung durch die Nutzung von in eine Datenbank eingepflegte Dienste (real existierende Dienste) erweitert werden.

Vorbedingung:

- Es wurde ein Konzept zur Einbettung von Diensten und Dienstleistungen in kontextuelle Bookmarks entwickelt und in einer Präsentation vorgestellt.

Nachbedingung:

- Die nun in der Datenbank vorhandenen Dienste werden durch den Client in einer Live-Demo abgefragt und ihre Realisierung in einer Präsentation vorgestellt.

Tasks	Verantw.	Erl.	S.G.	S.V.
Auf real existierende Dienste anpassen	Christan M. & Torben	X	11	12
Erstellung einer Präsentation zu real existierenden Diensten	Christan M. & Torben	-	-	-

Projekt-Dokumentation

Lfd-Nummer: 1	Priorität: 0	
Projekt-Dokumentation		
Beginn: 01.10.2007	Risiko: niedrig	Zeit: bis Projektende
BESCHREIBUNG		
Die Dokumentation des Projektes wird durch die Verwendung von XP iterativ ergänzt und nach Abschluss des Projektes müssen die verschiedenen Aspekte der Projektarbeit in einer Dokumentation nachlesbar sein. Diese wird online einsehbar sein und die Projekt-Homepage bietet einen aktuellen und optisch ansprechenden Überblick über den Status der Projektarbeit. Der Projekt-Endbericht wird fortlaufend iterativ erweitert.		
TASKS	VERANTWORTLICH	ERLEDIGT
Klassendiagramm als rückblickende Übersicht	Jan	X
Grundgerüst des Endberichts	Christian M.	X
Benutzerhandbuch	-	-
Online-Hilfe	-	-
Umstrukturierung des Endberichts	Marcel	X
Erfassung der Daten, die auf die Website gehören	Dennis	X

Website aktualisieren

<table>
<tr><td colspan="2">Lfd-Nummer: 15</td><td>Priorität: 0</td></tr>
<tr><td colspan="3">Website aktualisieren</td></tr>
<tr><td>Beginn: 04.03.2008</td><td>Risiko: niedrig</td><td>Zeit: bis Projektende</td></tr>
<tr><td colspan="3">BESCHREIBUNG</td></tr>
<tr><td colspan="3">Die Website eines Projektes dient als Aushängeschild und Repräsentation in der Öffentlichkeit und stellt den aktuellen Stand einer Projektarbeit dar. Durch die ständige Weiterentwicklung der Anwendung und damit verbundener Erreichung neuer Releases und Iterationen ist es notwendig, die Daten der Website aktuell zu halten. Die Website bietet den aktuellen Entwicklungsstand der Anwendung sowie eine Darstellung des Projektes (Beschreibung, Zielsetzung, Vorgehensweise und aktuellen Entwicklungsstand) selbst.</td></tr>
<tr><td colspan="3">Vorbedingung:
• Keine Vorbedingungen.
Nachbedingung:
• Die Webseite zeigt den aktuellen Stand der Entwicklung, eine Beschreibung und die Zielsetzung des Projektes.</td></tr>
<tr><td>TASKS</td><td>VERANTWORTLICH</td><td>ERLEDIGT</td></tr>
<tr><td>Erfassung der Daten, die auf die Website gehören</td><td>Dennis</td><td>-</td></tr>
<tr><td>Stand des Projektes auf der Homepage grafisch darstellen</td><td>Dennis</td><td>-</td></tr>
</table>

Erkennung von Kartenausschnitten zu Navigationszwecken

Lfd-Nummer: 22	Priorität: 0			
Erkennung von Kartenausschnitten zu Navigationszwecken				
Beginn: 19.06.2008	Risiko: mittel	Zeit: n.a.		
Beschreibung				
Momentan liegt der Projektfokus auf den Bereich der Plakaterkennung. Denkbar ist jedoch eine weitere Nutzung der vorhandenen SIFT-Implementierung für andere Bereiche. Als Beispiel dient hierbei die Erkennung von Kartenausschnitten zur Navigation zu naheliegenden Events (Beispiel: Heideparkplan). Hierzu muss zunächst geprüft werden, ob die existierende SIFT-Implementierung dazu geeignet ist. Anschließend müssen geeignete Services für Kartenausschnitte definiert werden.				
Vorbedingung: • Keine Vorbedingungen. *Nachbedingung:* • Keine Nachbedingungen.				
Tasks	Verantw.	Erl.	S.g.	S.v.

10.4.2. Umsetzung

Im Folgenden werden die Details der in Iteration II implementierten Softwarekomponenten und die verwendeten Strukturen kurz erläutert.

Datenstruktur

Um eine verbesserte Personalisierung zu erreichen, wurde die Datenstruktur überarbeitet und erweitert. Zunächst wurde als Beispiel ein fiktiver Benutzer angenommen, der ein Kinoplakat fotografiert und dazu Dienste abrufen möchte. In seinem Benutzerprofil ist hinterlegt, dass er mexikanisches Essen mag und Raucher ist.

Fotografiert dieser Benutzer ein Kinoplakat in Oldenburg, soll er Informationen zu nahegelegenen Kinos, in denen dieser Film läuft, bekommen. Auch soll er die Möglichkeit haben, Karten zu einer Vorstellung zu kaufen. Läuft dieser Film nicht mehr im Kino, soll der Benutzer diesen Film auf DVD vorbestellen können oder eine Erinnerung in seinen Kalender setzen, die eine entsprechende Meldung ausgibt, sobald der Film auf DVD erschienen ist. Gibt es den Film schon auf DVD, sollen nahe gelegene Händler angezeigt

werden, die diese anbieten, oder Online-Bestellmöglichkeiten gegeben werden. Eine weitere Option wäre, eine Einladung per E-Mail an beliebige Einträge aus dem Adressbuch zu senden für einen Videoabend mit diesem Film.

Aufgrund seines Profils soll der fiktive Nutzer die Option erhalten, nach der Kinovorstellung einen Tisch im nächstgelegenen mexikanischen Restaurant zu reservieren. Restaurants, in denen geraucht werden darf, werden in diesem Fall bevorzugt.

Anhand des genannten Beispiels wurde überlegt, unter welchen Umständen welche Daten an den verschiedenen Stellen des Systems benötigt werden. Auf jeden Fall wird die *Zeit*, zu der ein Benutzer das Plakat fotografiert und dessen *Ort* benötigt. Es sollen beispielsweise keine Karten zu vergangenen Kinovorstellungen gekauft werden können. Es sollen auch keine Vorstellungen von weiter entfernten Kinos angeboten werden, es sei denn, der Benutzer hat in seinem *Benutzerprofil* diesen Ort als Heimatort angegeben oder er wählt explizit diesen Ort für eine Vorstellung aus. Der Ort des Benutzers kann auf dem Handy durch Abfragen der GPS-Daten ermittelt werden. Um aus den Koordinaten einen Ort zu ermitteln, wurde ein Reverse Geocoding eingeführt. Die Koordinaten werden an einen entsprechenden Server im Internet gesendet und dieser liefert den zugehörigen Ort zurück.

Dienste

Die Dienste, die auf dem Handy dargestellt werden, lassen sich auf einige wenige, immer wiederkehrende Diensttypen reduzieren:

- Informationen anzeigen (z. B. Route, Produktinfo)
- Platz reservieren, evtl. durch Kartenkauf (z. B. Kino, Konzert, Tisch im Restaurant)
- Etwas kaufen (z. B. DVDs, CDs, Pizza-Service)
- Erinnerung setzen (in Kalender)
- Informationen versenden (per E-Mail, SMS)

Es wurde überlegt, ob sich durch diese Regelmäßigkeit die Implementierungsarbeit im Handy-Client vereinfachen ließe. Für den Fall, dass die Dienste im Client innerhalb der Anwendung angezeigt werden sollten (anstatt als Webseite im Browser), müsste die Darstellung für jeden Diensttyp entsprechend implementiert werden.

Eine weitere Überlegung war, wie weit bei der Implementierung der Dienste ins Detail gegangen werden sollte: Wenn der Benutzer eine Kinokarte kaufen möchte, soll er auch den Kinosaal auswählen können, falls der Film in einem Kino in mehr als einem Saal läuft? Soll er auch einen Sitzplatz auswählen können?

Als mögliche Lösung wurde in Betracht gezogen, nur das zu implementieren, was auch direkt mit den kontextuellen Informationen zusammenhängt. Alles weitere könnte vom

Dienstanbieter über eine Art Formularbaukasten im Web-Frontend eingepflegt werden. Damit könnte der Dienstanbieter verschiedene Formularelemente (Dropdown-Menüs, Radio-Buttons etc.) einsetzen um Benutzereingaben zu erhalten.

Diese Lösung wurde aber verworfen, stattdessen sollte nur das eingebaut werden, was unmittelbar mit kontextuellen Informationen zusammenhängt. Für weitere Informationen wie Kinosaal und Sitzplatz wurde eine Weiterleitung auf die Webseite des Anbieters angedacht. Der genaue Ablauf wurde folgendermaßen festgelegt: Wird ein Foto vom Server korrekt erkannt, erhält der Benutzer eine Liste mit zugehörigen Diensten. Wählt er einen der Dienste aus, wird eine Instanz des Handy-Browsers geöffnet und der Dienst als Webseite dort angezeigt.

Die Auswahl der Dienste, d.h. die Zusammenstellung der Dienstlisten anhand der Orts- und Zeitdaten sowie des Benutzerprofils erfolgt mittels der Suchmaschine *Lucene*. Diese wird mit geeigneten Daten versorgt und stellt daraufhin die Dienstlisten für den Client zusammen.

Datenstruktur

Die Datenstruktur (Abb. 10.4) wurde etwas erweitert, um die Orte, Benutzerprofile, Diensttypen und -zeiten in der Datenbank zu speichern.

Die Speicherung der Zeiten, zu denen ein Dienst gültig sein soll, musste recht feingranular erfolgen. Ein Dienst kann zu einem festen Zeitpunkt gültig sein, er kann innerhalb eines Zeitraums gültig sein, sowie zu immer wiederkehrenden Zeitpunkten oder Zeiträumen gültig sein. Geplant waren hierfür in der Diensttabelle die Einträge *date_from*, *date_duration*, *time_from*, *time_duration*, *repeat_time_type* und *repeat_value*.

Es wurde entsprechend des Prinzips „Simplicity“ jedoch beschlossen, diese recht aufwändige Struktur nicht umzusetzen und stattdessen sehr einfache Einträge der Form *timestamp_from*, *timestamp_to* zu nutzen. Bei wiederkehrenden Diensten oder Diensten, die nur kurzzeitig nicht verfügbar sind, sollten mehrere dieser einfachen Einträge die kompliziertere Struktur ersetzen.

Hauptargumente für den vereinfachten Aufbau waren, dass mit weniger komplizierten Einträgen der Rechenaufwand pro Eintrag geringer sei, wodurch sich das Vorhandensein mehrerer Einträge ausgleichen würde, und dass die Umsetzung deutlich einfacher und damit auch weniger fehleranfällig sei.

Frontend

Das begonnene Web-Frontend musste weiter ausgebaut werden, um der neuen Datenstruktur zu entsprechen: Anbieter wählen den gewünschten Diensttyp und geben einen oder mehrere Orte sowie die Zeiten an, zu denen der Dienst verfügbar sein soll. Dabei werden dann Plakate mit Zeiten und Orten zu Diensten verknüpft.

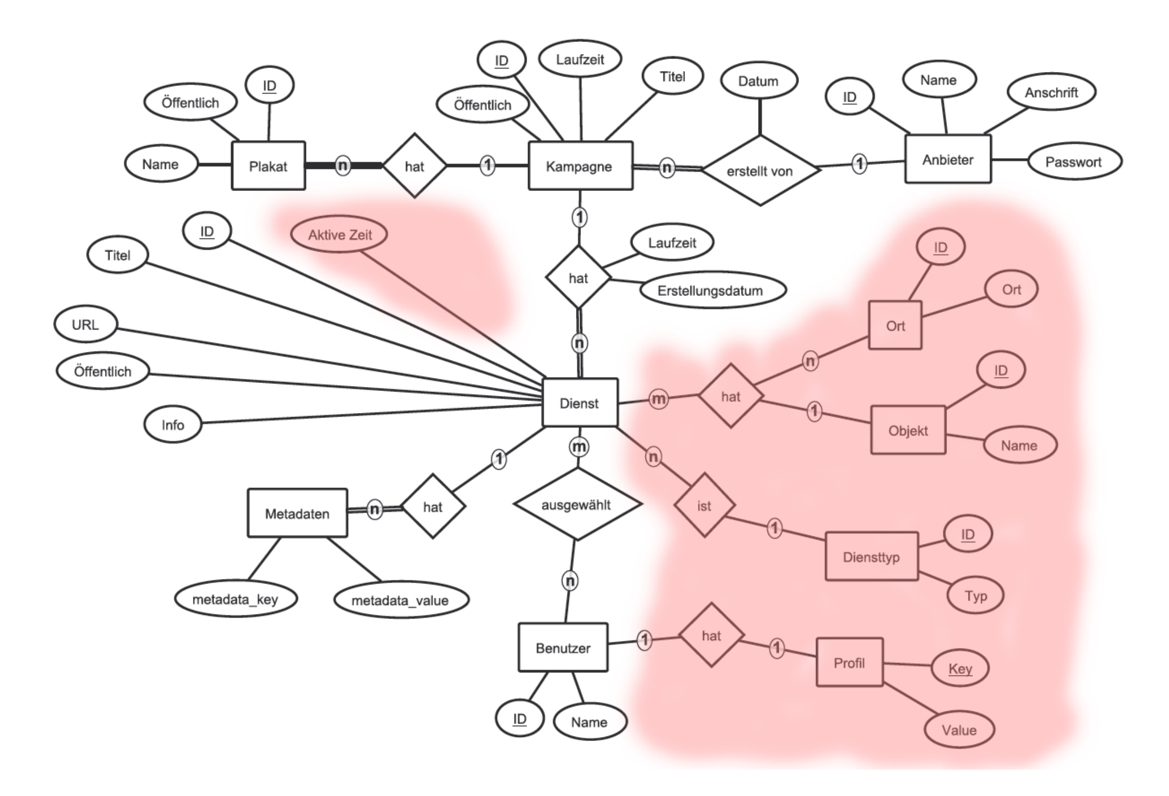


Abbildung 10.4.: ER-Diagramm der Datenbankstruktur nach der Personalisierung

Aus Zeitgründen wurde aber beschlossen, die Änderungen des Frontends erst in einer späteren Iteration stattfinden zu lassen.

Systemarchitektur

Die Architektur des Clients blieb wie in der letzten Iteration unverändert. Durch den Fortschritt des Projektes und die neuen Funktionalitäten des Servers in Bezug auf Datenspeicherung und kontextuelle Bookmarks musste dort eine Umstrukturierung der Pakete stattfinden.

Die einzelnen Pakete des Servers wurden im Laufe der Iteration nach logischer Zusammengehörigkeit und Funktionalität gekapselt. Eine angepasste Darstellung der Serverarchitektur findet sich in Abbildung 10.5. Das Paket „converter“ beeinhaltet alle Klassen zur XML-Verarbeitung, „beans“ das Datenmodell der Anwendung, „contextual“ die Elemente zur Verarbeitung und Erstellung von kontextuellen Bookmarks, „persistence“ die Datenbankanbindung, „controller und network“ die Klassen zur Client/Server Kommunikation, „action“ die Elemnte zum Webzugriff auf den Server (Serverstatus und Reboot) und „util“ einige Hilfsklassen (wie eigens für die Anwendung definierte Exceptions).

10.4.3. Ergebnisse

Ergebnisse der Akzeptanztests

In dieser Iteration wurden alle Storycards mit der Priorität 1 und 2 behandelt und die dementsprechenden Akzeptanztests bearbeitet. Die Implementierung der Personalisierung und der kontextuellen Bookmarks wurde anhand eines Beispiels demonstriert, in dem die Personalisierung durch zwei fiktive Nutzer - einem Vegetarier und einem Carnivoren - dargestellt wurde (vgl. 10.4.1 und 10.4.1).

Das Konzept zur aktuellen Evaluation (vgl. 10.4.1) wurde präsentiert, jedoch stand die Präsentation der Durchführungsergebnisse zum Ende der Iteration noch aus.

Die Anpassung der Client-GUI (vgl. 10.4.1) wurde aufgrund der neuen Evaluation zurückgestellt und zusammen mit allen Storycards einer höheren Priorität in die nächste Iteration übernommen.

Retrospektive

In dieser Iteration wurde damit begonnen, zu jeder Gruppensitzung eine Präsentation zu erstellen. Dies wurde sowohl von den Betreuern, als auch von allen Gruppenmitgliedern sehr positiv aufgenommen, da damit sehr transparentes Arbeiten möglich wurde.

Allerdings zeigte sich in dieser Iteration auch, dass der Server-Code langsam immer komplexer wurde, wodurch es immer schwieriger wurde, diesen zu verstehen. Wenn ein neues Feature implementiert werden sollte, wurde oft unerwartet viel Zeit dafür benötigt, da stets zunächst der Server-Code neu durchdrungen werden musste. Daher wurden umfangreiche Refactoring-Maßnahmen im Server durchgeführt, so dass dieses Problem nach Abschluss der Iteration nicht mehr bestand.

Als größere Teile der Besprechungen in der Iteration aufgrund eines pakistanischen Gastes auf englisch gehalten wurden, zeigte sich, dass dort eine Diskussion nicht richtig zustande kam, da die Gruppenmitglieder nicht sehr geübt in der englischen Sprache waren und sich viele daher im Vergleich zu den anderen Treffen stärker zurückhielten. Durch eine rechtzeitige Ankündigung, dass die Gruppensitzung auf englisch stattfinden würde, wäre eine bessere Vorbereitung und insbesondere das Erstellen der Präsentationsfolien auf englisch möglich gewesen.

Weiterhin zeigte sich, dass die Durchführung von einigermaßen realistischen Aufwandseinschätzungen allgemein schwierig ist. Vor allem der Versuch, die Software mit UMTS auf dem Handy zu nutzen, nahm aufgrund unerwarteter Probleme viel mehr Stunden in Anspruch als ursprünglich geplant.

10.5. Iteration III

Die dritte und letzte Iteration von Release 3 erstreckte sich über die Kalenderwochen 28 bis 31 und lag damit im Monat Juli. Auf Grund der Klausurphase und der beginnenden Ferien war die Projektgruppe personell geschwächt.

Es sollten in der Iteration lediglich die Tasks aus den letzten Iterationen weiter bearbeitet werden, neue Storycards wurden nicht hinzugefügt. Da das Ende der Projektgruppe nach Abschluss dieser Iteration nur noch etwa zwei Monate entfernt liegen würde, wurde mit Priorität daran gearbeitet, die grundlegende Funktionalität des Gesamtsystems möglichst bald fertigzustellen.

10.5.1. Storycards

In der Iteration wurden die im Nachfolgenden aufgelisteten Storycards behandelt.

Ströer Medien - Evaluation

Lfd-Nummer: 21	Priorität: 1	
Ströer Medien - Evaluation		
Beginn: 19.06.2008	Risiko: mittel	Zeit: 153 h

Beschreibung

Im Kontext einer Kooperation mit der Firma Ströer Medien im Rahmen einer weiteren Veranstaltung wird Anfang Juli 2008 eine Plakatkampagne gestartet. Diese bietet die Möglichkeit, unsere Anwendung mit Hilfe von universitätsexternen Personen zu evaluieren. War bei der Evaluation 10.3.1 der Vergleich zwischen herkömmlichem Internetbrowser und dem „Point and Shoot "-Verfahren von Interesse, soll in diesem Rahmen nach weiteren Evaluationsmöglichkeiten gesucht werden, ein Konzept erstellt und dieses durchgeführt werden.

Vorbedingung:

- Keine Vorbedingungen.

Nachbedingung:

- Ein Konzept zur Evaluation wurde erstellt und in einer Präsentation vorgestellt.
- Die Evaluation wurde durchgeführt und die Ergebnisse in einer Präsentation vorgestellt.

Tasks	Verantw.	Erl.	S.G.	S.V.
Überprüfung der Handyfunktionalität auf UMTS	Jeschua & Sören	-	6	-
Erstellung eines Konzeptes zur Evaluation	Christian N. & Sören	X	37	36
Ausarbeitung eines Fragebogens	Christian N. & Sören & Marcel	X	20	22
Durchführung der Evaluation	Christian N. & Sören & Marcel & Jeschua	X	60	28
Auswertung der Evaluation und Erstellung einer Präsentation	Christian N. & Sören & Marcel & Jan	X	30	-

Client GUI anpassen

Lfd-Nummer: 10	Priorität: 1	
Client GUI anpassen		
Beginn: 04.03.2008	Risiko: niedrig	Zeit: 4 h

Beschreibung
Die Attraktivität einer Anwendung für einen Benutzer steigt durch eine intuitive Bedienung, sowie einer optisch ansprechende Gestaltung. Durch ständige Erweiterungen und Strukturumstellungen muss der Handy-Client fortlaufend angepasst werden. Die grafische Oberfläche des Handy-Clients wird konsequent einen einheitlichen Gestaltungsstil verwenden und ansprechend gestaltet sein. Alle Elemente des UIs werden nach demselben Stil aufgebaut, gestaltet und bedienbar sein. Der optische Aufbau wird sich an anerkannte Standards des Software Engineerings halten. So können alle Funktionalitäten innerhalb von maximal drei bis vier Schritten erreicht werden.

Vorbedingung:

- Keine Vorbedingungen.

Nachbedingung:

- Der Handy-Client besitzt einen einheitlichen Gestaltungsstil.
- Alle Anwendungsfunktionen sind in maximal drei bis vier Schritten erreichbar.
- Die überarbeiteten Hilfe wurde in einer Live-Demo vorgestellt.

Tasks	Verantw.	Erl.	S.g.	S.v.
Ladebalken umgestalten (keine Nummern mehr anzeigen, „Sanduhr“)	Christian M. & Christian N.	X	n.A.	n.A.
Einen Entwurf zur grafischen Gestaltung des Handy-Clients erstellen	Jeschua	X	n.A.	n.A.
Die Elemente der Handy-GUI auf eine einheitliche Gestaltung überprüfen	Jeschua	X	n.A.	n.A.
Die Elemente der Handy-GUI an eine einheitlich Gestaltung anpassen	Jeschua	X	n.A.	n.A.
Skalierung der aufgenommenen Fotos auf die Größe des verwendeten Handy-Displays	Christian M. & Christian N.	X	n.A.	n.A.
GUI verschnellern, Speicherverbrauch minimieren	alle	X	n.A.	n.A.

Lfd-Nummer: 10	Priorität:			
Client GUI anpassen (Forts.)				
TASKS	VERANTW.	ERL.	S.G.	S.V.
Refactoring allgemein und speziell Camera-MIDlet	alle	X	n.A.	n.A.
J2ME-Polish auf neue Version updaten	Christian N. & Jeschua	X	n.A.	n.A.
Hilfe-Bug reparieren	Jeschua, Jan	X	n.A.	n.A.
IP-Eingabe überarbeiten	Chritian N. & Jeschua	X	5	5
Bug (Thread-Exception) in „Bild versenden“ beheben (tritt auf, wenn man zurück will)	Oliver & Jan	X	n.A.	n.A.
Zoom-Funktionalität	Oliver & Jan	X	15	22
Die Handy-Hilfemenüs auf den aktuellen Stand bringen (u.a. 90° Drehung des Handys)	Christian M.	X	4	2

Implementierung kontextueller Bookmarks

Lfd-Nummer: 13b	Priorität: 2	
Implementierung kontextueller Bookmarks		
Beginn: 04.03.2008	Risiko: hoch	Zeit: 51 h

Beschreibung
In Storycard 13a wurde ein Konzept zur Darstellung und Umsetzung von kontextuellen Bookmarks entworfen. Ziel ist es nun, dieses Konzept in der bestehenden Anwendung erfolgreich umzusetzen.

Vorbedingung:

- Es wurde ein Konzept zum Aufbau kontextueller Bookmarks erstellt und in einer Präsentation vorgestellt.

Nachbedingung:

- Die Umsetzung des Konzeptes wird mit Hilfe einer Präsentation und Demonstration mit Hilfe der Anwendung dargestellt.
- Die Demonstration läuft folgendermaßen ab: Der Benutzer nimmt ein Foto auf und sendet dieses an den Server. Dieser ermittelt den Standort des Benutzers und gibt kontextbezogene Bookmarks zu dem erkannten Foto zurück. Dieser Vorgang ist von Benutzern mit unterschiedlichem Profil zu wiederholen, damit der kontextuelle Hintergrund gezeigt werden kann.

Tasks	Verantw.	Erl.	S.G.	S.V.
Implementierung einer Datenstruktur für Bookmarks	Christian M. & Jeschua & Oliver & Torben	X	n.A.	n.A.
Laden/Speichern von kontextuellen Bookmarks auf dem Handy	Sören & Oliver	X	n.A.	n.A.
Zuvor angepasstes Konzept von kontextuellen Bookmarks implementieren	Dennis & Torben & Jan & Oliver	-	51	41 + Torben + Oliver

Frontend für kontextuelle Bookmarks

Lfd-Nummer: 19	Priorität: 3	
Frontend für kontextuelle Bookmarks		
Beginn: 12.05.2008	Risiko: niedrig	Zeit: 68 h

Beschreibung

Dem Kunden der Anwendung ist es möglich, seine Plakatdaten in das bestehende System einzupflegen. Daher wird eine grafisch ansprechende und zu allen Funktionen in drei bis vier Schritten erreichbare Anwendung entworfen und implementiert werden. Der Kunde kann reine Bilddaten, dazu passende Services und Metadaten für seine Plakatkampagne über das Frontend einstellen. Diese sind dann automatisch für jeden Client verfügbar.

Vorbedingung:

- Die endgültige Datenstruktur für kontextuelle Bookmarks muss festgelegt sein.
- Frontend muss Zugriff auf die Datenbank des Olivanda-Projektes besitzen.

Nachbedingung:

- Ein Plakatdatensatz kann in das aktiv laufende System mit Hilfe des Frontends eingebunden werden. Hierzu zählen Bilddaten, kontextuelle Bookmarks sowie genutzte Dienste des Plakaterstellers.
- Die Ergebnisse werden in einer Präsentation und in einer Live-Demo vorgestellt. Die durch das Frontend beispielhaft eingestellte Plakatkampagne wird über den Handy-Client durch Aufnahme des entsprechenden Plakates abgerufen und mit den entsprechenden Bookmarks dargestellt.

Tasks	Verantw.	Erl.	S.G.	S.V.
Recherche nach Realisierungsmöglichkeiten und bestehenden Technologien	-	X	7	4
Technologieauswahl	-	X	9	4
Konzept einer Datenstruktur	-	-	16	11
Konzept der Benutzeroberfläche	-	-	10	19
Implementierung der Datenstruktur	-	-	34	8
Implementierung des Frontends	-	-	26	32
Methoden der Datenbankschnittstelle ausfüllen	-	-	4	7
GUI vervollständigen und ausbauen	-	-	30	10
Besitzer zu Objekten hinzufügen	-	-	5	-

Unbekannte Objekte annotieren

Lfd-Nummer: 20	Priorität: 4	
Unbekannte Objekte annotieren		
Beginn: 12.05.2008	Risiko: mittel	Zeit: 29 h

BESCHREIBUNG
Bisher erhält der Benutzer als Rückmeldung, wenn ein gesendetes Foto zu keinem Plakat zugeordnet werden konnte, lediglich eine diesbezügliche Hinweismeldung. Ziel dieser Storycard ist es, dem Benutzer die Möglichkeit zu geben, Fotos zu annotieren. Hierzu soll er Kampagnen zum Foto auswählen, Dienste zuordnen und das Foto mit Metadaten anreichern können. Diese Daten kann er anschließend persistent in der Datenbank und für andere Nutzer zugänglich speichern.

Vorbedingung:

- Datenstruktur zur Speicherung von Plakaten und Bookmarks muss implementiert worden sein.
- Ein Konzept zur Speicherung und Annotation von zuvor unbekannten Objekten wurde erstellt und in einer Präsentation vorgestellt.

Nachbedingung:

- Das Konzept wurde implementiert und in einer Live-Demo vorgestellt.

TASKS	VERANTW.	ERL.	S.G.	S.V.
Konzepterstellung für die Annotation unbekannter Objekte und Präsentation	Oliver & Jan	X	5	7
Implementierung des Konzeptes zur Annotation unbekannter Objekte	Oliver & Jan	-	44	15

Evaluation der Algorithmen

Lfd-Nummer: 8	Priorität: 4	
Evaluation der Algorithmen		
Beginn: 19.02.2008	Risiko: niedrig	Zeit: 37 h

Beschreibung
Die Algorithmen werden anhand von verschiedenen Gesichtspunkten, wie Geschwindigkeit, Treffergenauigkeit und Robustheit einheitlich getestet und bewertet. Dies geschieht mit Hilfe eines festen Testmusters, welches ein Bilderset und fixe Grenzwerte beinhaltet.

Vorbedingung:

- Die Evaluation wurde mit einem geeignetem Testset (verschiedene Auflösungen und jeweils verschiedene Bildqualitäten) durchgeführt.

Nachbedingung:

- Die Evaluation wurde durchgeführt und ausführlich inkl. verwendeter Tools und entsprechenden Vergleichsbedingungen im Endbericht dargestellt.
- Die Funktionsfähigkeit der Algorithmenevaluation wird anhand der Evaluation eines exemplarischen Testsets demonstriert.

Tasks	Verantw.	Erl.	S.G.	S.V.
Überlegen eines Grobkonzeptes für die Evaluation	Marcel & Sören	X	n.A.	n.A.
Refactoring des Konzepts mit dem Betreuer	Marcel & Sören	X	n.A.	n.A.
Referanzplakate organisieren	Marcel & Sören	X	n.A.	n.A.
Referenzbilder aufbereiten	Marcel & Sören	X	n.A.	n.A.
Testsets zusammenstellen	Marcel & Sören	X	n.A.	n.A.
Algorithmentest implementieren	Marcel & Sören	X	n.A.	n.A.
Auswertung erstellen	Marcel & Sören	X	2	n.A.
Evaluation der Aufnahmeauflösung	-	-	n.A.	n.A.
Evaluationsauswertung entwickeln	Marcel & Dennis	X	11	12
Evaluation durchführen, Präsentation erarbeiten und vorstellen	Marcel & Dennis	X	8	4
Testsets erstellen (verschiedene Auflösungen, ähnliche Plakate)	-	-	20	-
Refactoring der Evaluationsimplementierung	Dennis	X	20	22
Evaluation erneut durchführen und Präsentation erstellen	-	-	11	-

Dienste in Bookmarks implementieren

Lfd-Nummer: 14b	Priorität:	
Dienste in Bookmarks implementieren		
Beginn: 04.03.2008	Risiko: hoch	Zeit: 11 h

Beschreibung
In Storycard 13b wurden Konzepte zur Realisierung von Diensten in Bookmarks erstellt und mit fest einprogrammierten Diensten implementiert. Nun soll diese Implementierung durch die Nutzung von in eine Datenbank eingepflegte Dienste (real existierende Dienste) erweitert werden.

Vorbedingung:

- Es wurde ein Konzept zur Einbettung von Diensten und Dienstleistungen in kontextuelle Bookmarks entwickelt und in einer Präsentation vorgestellt.

Nachbedingung:

- Die nun in der Datenbank vorhandenen Dienste werden durch den Client in einer Live-Demo abgefragt und ihre Realisierung in einer Präsentation vorgestellt.

Tasks	Verantw.	Erl.	S.g.	S.v.
Auf real existierende Dienste anpassen	Christan M. & Torben	X	11	12
Erstellung einer Präsentation zu real existierenden Diensten	Christan M. & Torben	-	-	-

Personalisierung implementieren

Lfd-Nummer: 17b		Priorität: 2
Personalisierung implementieren		
Beginn: 31.3.2008	Risiko: hoch	Zeit: 88 h

BESCHREIBUNG
In SC 10.3.1 wurde ein Konzept zur Personaliserung von Bookmarks erarbeitet. Dieses wird nun anhand der dort getroffenen Designentscheidungen implementiert.

Vorbedingung:

- Eine „state-of-the-art“ Analyse wurde durchgeführt, ein Verfahren wurde darauf basierend ausgewählt und in einer Präsentation vorgestellt.

Nachbedingung:

- Die Implementierung der Personalisierung wurde anhand einer Live-Demo und in einer Präsentation vorgestellt.
- Die Live-Demo ist äquivalent zu dem Ablauf des Akzeptanztests der SC 10.5.1.

TASKS	VERANTW.	ERL.	S.G.	S.V.
Implementierung der Personalisierung	Christian M. & Dennis & Torben		118	30
Erstellung einer Präsentation zur Personalisierung	Christian M. & Dennis & Torben	X	-	3

Textdetektion und Texterkennung integrieren

Lfd-Nummer: 18b	Priorität: 4	
Textdetektion und Texterkennung implementieren		
Beginn: 12.05.2008	Risiko: hoch	Zeit: 54 h

BESCHREIBUNG
In Storycard 18a wurden verschiedene Alternativen und Ergänzungen zu der bereits im Projekt realisierten SIFT-Variante diskutiert. Als weitere Vorgehensweise wurde beschlossen, Textdetektion und Texterkennung in das Projekt zu integrieren.

Vorbedingung:

- Alternative Techniken wurden recherchiert und diskutiert

Nachbedingung:

- Es wurde ein Konzept zur Kombination von SIFT mit Textdetektion und Texterkennung in einer Präsentation vorgestellt.
- Texterkennung und Detektion wurden implementiert und in einer Live-Demo vorgestellt, diese umfasst alle im Konzept definierten Funktionalitäten.

TASKS	VERANTW.	ERL.	S.G.	S.V.
Auswahl des zu nutzenden Frameworks für die Textdetektion und Texterkennung	Torben & Jan	-	4	-
Erstellung eines Konzeptes zur Integration von Textdetektion und Texterkennung in die bereits bestehende SIFT-Architektur erarbeiten und eine Präsentation erstellen	-	-	-	-
Implementierung von Textdetektion und Texterkennung im Projekt	Torben & Jan	-	50	-

Erkennung von Kartenausschnitten zu Navigationszwecken

Lfd-Nummer: 22	Priorität: 0	
Erkennung von Kartenausschnitten zu Navigationszwecken		
Beginn: 19.06.2008	Risiko: mittel	Zeit: n.a.

BESCHREIBUNG
Momentan liegt der Projektfokus auf den Bereich der Plakaterkennung. Denkbar ist jedoch eine weitere Nutzung der vorhandenen SIFT-Implementierung für andere Bereiche. Als Beispiel dient hierbei die Erkennung von Kartenausschnitten zur Navigation zu naheliegenden Events (Beispiel: Heideparkplan). Hierzu muss zunächst geprüft werden, ob die existierende SIFT-Implementierung dazu geeignet ist. Anschließend müssen geeignete Services für Kartenausschnitte definiert werden.

Vorbedingung:

- Keine Vorbedingungen.

Nachbedingung:

- Ein Konzept zur Nutzung und Darstellung zur Erkennung von Kartenausschnitten wurde erstellt und in einer Präsentation vorgestellt.
- Der Algorithmentest wurde mit einem Testset an Kartenausschnitten in einer Live-Demo und anhand einer Ergebnispräsentation vorgestellt.

TASKS	VERANTW.	ERL.	S.G.	S.V.
Konzept zur Nutzung und Darstellung	Dennis	X	30	15
Evaluation von Kartenerkennung mit Algorithmentest	Dennis	X	10	7

Projekt-Dokumentation

<table>
<tr><td>Lfd-Nummer: 1</td><td colspan="2">Priorität: 0</td></tr>
<tr><td colspan="3">Projekt-Dokumentation</td></tr>
<tr><td>Beginn: 01.10.2007</td><td>Risiko: niedrig</td><td>Zeit: bis Projektende</td></tr>
<tr><td colspan="3">Beschreibung</td></tr>
<tr><td colspan="3">Die Dokumentation des Projektes wird durch die Verwendung von XP iterativ ergänzt und nach Abschluss des Projektes müssen die verschiedenen Aspekte der Projektarbeit in einer Dokumentation nachlesbar sein. Diese wird online einsehbar sein und die Projekt-Homepage bietet einen aktuellen und optisch ansprechenden Überblick über den Status der Projektarbeit. Der Projekt-Endbericht wird fortlaufend iterativ erweitert.</td></tr>
<tr><td>Tasks</td><td>Verantwortlich</td><td>Erledigt</td></tr>
<tr><td>Klassendiagramm als rückblickende Übersicht</td><td>Jan</td><td>X</td></tr>
<tr><td>Grundgerüst des Endberichts</td><td>Christian M.</td><td>X</td></tr>
<tr><td>Benutzerhandbuch</td><td>-</td><td>-</td></tr>
<tr><td>Online-Hilfe</td><td>-</td><td>-</td></tr>
<tr><td>Umstrukturierung des Endberichts</td><td>Marcel</td><td>X</td></tr>
<tr><td>Erfassung der Daten, die auf die Website gehören</td><td>Dennis</td><td>X</td></tr>
</table>

Website aktualisieren

<table>
<tr><td colspan="2">Lfd-Nummer: 15</td><td>Priorität: 0</td></tr>
<tr><td colspan="3">Website aktualisieren</td></tr>
<tr><td>Beginn: 04.03.2008</td><td>Risiko: niedrig</td><td>Zeit: bis Projektende</td></tr>
<tr><td colspan="3">Beschreibung</td></tr>
<tr><td colspan="3">Die Website eines Projektes dient als Aushängeschild und Repräsentation in der Öffentlichkeit und stellt den aktuellen Stand einer Projektarbeit dar. Durch die ständige Weiterentwicklung der Anwendung und damit verbundener Erreichung neuer Releases und Iterationen ist es notwendig, die Daten der Website aktuell zu halten. Die Website bietet den aktuellen Entwicklungsstand der Anwendung sowie eine Darstellung des Projektes (Beschreibung, Zielsetzung, Vorgehensweise und aktuellen Entwicklungsstand) selbst.</td></tr>
<tr><td colspan="3">Vorbedingung:
• Keine Vorbedingungen.
Nachbedingung:
• Die Webseite zeigt den aktuellen Stand der Entwicklung, eine Beschreibung und die Zielsetzung des Projektes.</td></tr>
<tr><td>Tasks</td><td>Verantwortlich</td><td>Erledigt</td></tr>
<tr><td>Erfassung der Daten, die auf die Website gehören</td><td>Dennis</td><td>-</td></tr>
<tr><td>Stand des Projektes auf der Homepage grafisch darstellen</td><td>Dennis</td><td>-</td></tr>
</table>

10.5.2. Umsetzung

Betrachtete Algorithmen

Da primär die softwaretechnische Realisierung bereits erarbeiteter Konzepte anstand und insgesamt nur wenige Tasks bearbeitet werden konnten, wurden in dieser Iteration keine neuen Konzepte erarbeitet und daher auch keine neuen Verfahren oder Algorithmen in das Projekt eingebracht.

Systemarchitektur

Die Systemarchitektur wurde in dieser Iteration nicht weiter verändert und der unter 10.4.2 beschriebene Stand des Servers und des Handy-Clients blieben weiterhin aktuell.

10.5.3. Ergebnisse

Insgesamt konnten in dieser Iteration wegen der personellen Schwächung der Gruppe nur wenige Tasks bearbeitet werden. Somit lag der Schwerpunkt auf allein stehenden Tasks, die unabhängig vom Gesamtsystem bearbeitet werden konnten – beispielsweise das Anpassen der Client-Hilfe.

Zum Ende der Iteration wurde der Stand der Arbeiten an der Schaffung eines PHP-Frontends zur Erstellung von neuen Kampagnen (vgl. 10.5.1) präsentiert. Bis zur produktiven Funktionstüchtigkeit des Frontends waren allerdings noch weitere Arbeiten nötig, so dass die Storycard in die nächste Iteration übernommen wurde.

Ergebnisse der Akzeptanztests

Der Akzeptanztest zur Client-GUI wurde am Ende der Iteration mit Demonstration der aktualisierten Handy-Hilfe erfüllt (vgl. 10.5.1). Nach Vorstellung des Konzeptes zur Texterkennung wurde festgelegt, dass die anschließende Implementierung zugunsten höher priorisierter Storycards zurückgestellt werden sollte (vgl. 10.5.1).

Zum Ende der vorangegangenen Iteration war der Akzeptanztest zur Stöer-Evaluation (vgl. 10.5.1) noch nicht erfüllt worden, da die Präsentation der durch sie gewonnenen Ergebnisse noch ausstand. Die Ergebnisse wurden daher in dieser Iteration vorgestellt und die Storycard somit erfüllt. Als Fazit hierzu wurde festgehalten, dass die „Zurück“-Buttons des Handy-Clients überarbeitet werden sollten sowie die „Verschicken“-Abfrage herausgenommen werden sollte.

Da nicht mehr genügend Zeit bis zum Projektende sein würde, um alle noch ausstehenden Storycards und ihre Akzeptanstests zu erfüllen, musste für die nachfolgende Iteration eine Auswahl getroffen werden, welche Storycards noch umgesetzt werden sollten.

Retrospektive

Als dominierender Eindruck blieb in dieser Iteration die personelle Schwächung aufgrund der Klausurphase und anschließendem Urlaub vieler Mitglieder. So waren über zwei Wochen nur zwei Gruppenmitglieder anwesend, in den anderen beiden Wochen fehlten ebenfalls mehr als die Hälfte. Entsprechend gering waren die Fortschritte. Im Nachhinein wurde diskutiert, ob in dieser Zeit die Projektgruppe hätte ausgesetzt werden sollen.

Trotzdem bzw. gerade wegen der beginnenden vorlesungsfreien Zeit war die Stimmung in der Gruppe gut, zumal ein endgültiger Abschluss der Bearbeitung bei einigen Storycards absehbar wurde.

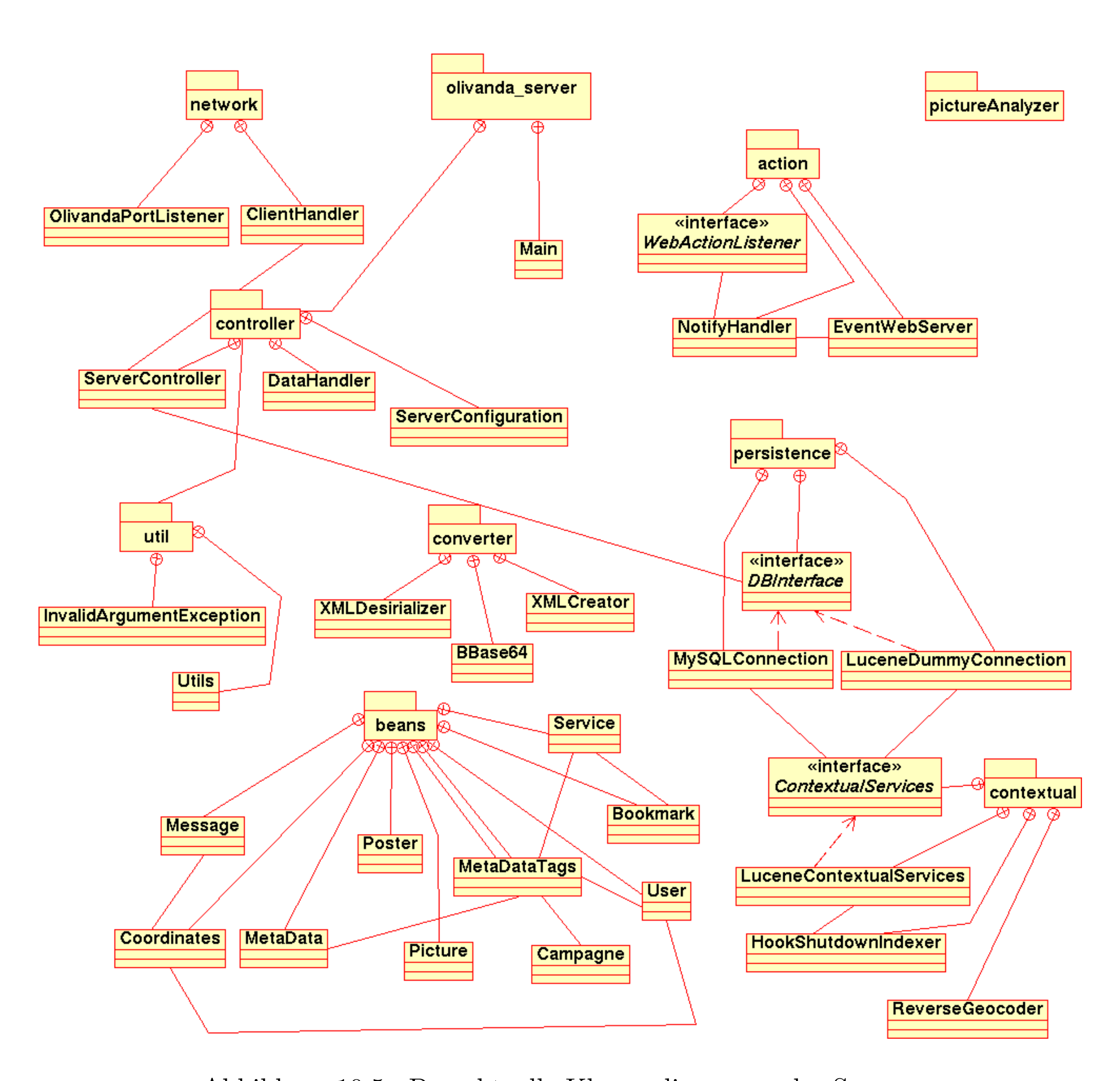


Abbildung 10.5.: Das aktuelle Klassendiagramm des Servers

11. Release IV (Zeitraum: 04.08.2008 bis 30.09.2008)

Das vierte Release stellte das letzte Release der Projektgruppe dar. Somit mussten sowohl die Arbeiten an den vorhandenen Storycards abgeschlossen werden, als auch abschließende Aufgaben wie das Packaging und die vollständige Dokumentation des Projekts durchgeführt werden.

11.1. Zielsetzung

Hauptziel war der Abschluss der essentiellen Storycards, beispielsweise der kontextuellen Bookmarks und des Webfrontends. Im Anschluss daran musste die Software „installationsfertig“ gemacht, sowie der Endbericht vervollständigt werden.

Einen wichtigen Anteil dieses Releases machte auch das Vorbereiten sowohl einer internen Abschlusspräsentation für die Abteilung Medieninformatik als auch der offiziellen Präsentation der Projektgruppe am Projektgruppentag des Departments für Informatik aus.

11.2. Zeitplan

Der derzeitige Stand innerhalb des globalen Projektzeitplanes (vgl. 1.6) zu Beginn von Release IV wird in Abb. 11.1 durch blau und lila hinterlegte Zeilen dargestellt.

- Iteration I:
  Start: 04.08.2008

- Iteration II:
  Start: 01.09.2008

Kalenderwoche	Release	Iteration	Woche
(01.01.2008)	1	1	1
3	1	2	1
4	1	3	1
5	1	4	1
(04.02.2008)	2	1	1
7	2	1	2
8	2	1	3
9	2	2	1
10	2	2	2
11	2	2	3
12	2	2	4
13	2	3	1
14	2	3	2
15	2	3	3
16	2	3	4
17	2	3	5
18	2	3	6
19	2	3	7
(12.05.2008)	3	1	1
21	3	1	2
22	3	1	3
23	3	1	4
24	3	2	1
25	3	2	2
26	3	2	3
27	3	2	4
28	3	3	1
29	3	3	2
30	3	3	3
31	3	3	4
(04.08.2008)	4	1	1
33	4	1	2
34	4	1	3
35	4	1	4
36	4	2	1
37	4	2	2
38	4	2	3
(30.09.2008)	4	2	4

Abbildung 11.1.: Zeitplanung Release IV

11.3. Iteration I

Die erste Iteration des letzten Releases fand in den Kalenderwochen 32-35 statt, wie aus Abbildung 11.1 zu entnehmen. Obwohl in der ersten Woche noch sehr viele Projektgruppenmitglieder im Urlaub waren, war die Projektgruppe in den verbleibenden drei Wochen fast voll besetzt. Es wurden Abschlussarbeiten an den peripheren Systemen wie dem Webfrontend, dem Annotationssystem und der Installations- und Konfigurationsoberfläche des Servers durchgeführt.

11.3.1. Storycards

Die in Iteration I eingesetzten Storycards bestehen weitestgehend aus den Storycards des Releases III, Iteration III. Es wurden nur die bereits abgeschlossenen Storycards entfernt. Besonderes Augenmerk lag auf den Storycards „Frontend für kontextuelle Bookmarks" (19) und „Unbekannte Objekte annotieren" (20). Desweiteren sollten die Storycards „Implementierung der Personalisierung" (17b) und „Implementierung kontextueller Bookmarks" (13b) abgeschlossen werden. Aus Zeitmangel wurden alle weiteren Storycards, abgesehen von der Projektdokumentation und der Aktualisierung der Website, nicht weiter bearbeitet.

Projekt-Dokumentation

Lfd-Nummer: 1	Priorität: 0	
Projekt-Dokumentation		
Beginn: 01.10.2007	Risiko: niedrig	Zeit: bis Projektende
BESCHREIBUNG		
Die Dokumentation des Projektes wird durch die Verwendung von XP iterativ ergänzt und nach Abschluss des Projektes müssen die verschiedenen Aspekte der Projektarbeit in einer Dokumentation nachlesbar sein. Diese wird online einsehbar sein und die Projekt-Homepage bietet einen aktuellen und optisch ansprechenden Überblick über den Status der Projektarbeit. Der Projekt-Endbericht wird fortlaufend iterativ erweitert.		
TASKS	VERANTWORTLICH	ERLEDIGT
Klassendiagramm als rückblickende Übersicht	Jan	X
Grundgerüst des Endberichts	Christian M.	X
Benutzerhandbuch	-	-
Online-Hilfe	-	-
Umstrukturierung des Endberichts	Marcel	X
Erfassung der Daten, die auf die Website gehören	Dennis	X

Implementierung kontextueller Bookmarks

Lfd-Nummer: 13b	Priorität: 9	
Implementierung kontextueller Bookmarks		
Beginn: 04.03.2008	Risiko: hoch	Zeit: 51 h

Beschreibung

In Storycard 13a wurde ein Konzept zur Darstellung und Umsetzung von kontextuellen Bookmarks entworfen. Ziel ist es nun, dieses Konzept in der bestehenden Anwendung erfolgreich umzusetzen.

Vorbedingung:

- Es wurde ein Konzept zum Aufbau kontextueller Bookmarks erstellt und in einer Präsentation vorgestellt.

Nachbedingung:

- Die Umsetzung des Konzeptes wird mit Hilfe einer Präsentation und Demonstration mit Hilfe der Anwendung dargestellt.
- Die Demonstration läuft folgendermaßen ab: Der Benutzer nimmt ein Foto auf und sendet dieses an den Server. Dieser ermittelt den Standort des Benutzers und gibt kontextbezogene Bookmarks zu dem erkannten Foto zurück. Dieser Vorgang ist von Benutzern mit unterschiedlichem Profil zu wiederholen, damit der kontextuelle Hintergrund gezeigt werden kann.

Tasks	Verantw.	Erl.	S.G.	S.V.
Implementierung einer Datenstruktur für Bookmarks	Christian M. & Jeschua & Oliver & Torben	X	n.A.	n.A.
Laden/Speichern von kontextuellen Bookmarks auf dem Handy	Oliver & Jan	X	n.A.	26
Zuvor angepasstes Konzept von kontextuellen Bookmarks implementieren	Dennis & Torben & Jan & Oliver	X	n.A.	61

Website aktualisieren

Lfd-Nummer: 15	Priorität: 0
Website aktualisieren	

Beginn: 04.03.2008	Risiko: niedrig	Zeit: bis Projektende

Beschreibung
Die Website eines Projektes dient als Aushängeschild und Repräsentation in der Öffentlichkeit und stellt den aktuellen Stand einer Projektarbeit dar. Durch die ständige Weiterentwicklung der Anwendung und damit verbundener Erreichung neuer Releases und Iterationen ist es notwendig, die Daten der Website aktuell zu halten. Die Website bietet den aktuellen Entwicklungsstand der Anwendung sowie eine Darstellung des Projektes (Beschreibung, Zielsetzung, Vorgehensweise und aktuellen Entwicklungsstand) selbst.

Vorbedingung:

- Keine Vorbedingungen.

Nachbedingung:

- Die Webseite zeigt den aktuellen Stand der Entwicklung, eine Beschreibung und die Zielsetzung des Projektes.

Tasks	Verantwortlich	Erledigt
Erfassung der Daten, die auf die Website gehören	Dennis	-
Stand des Projektes auf der Homepage grafisch darstellen	Dennis	-

Personalisierung implementieren

<table>
<tr><td colspan="3">Lfd-Nummer: 17b</td><td colspan="3">Priorität: 9</td></tr>
<tr><td colspan="6">Personalisierung implementieren</td></tr>
<tr><td>Beginn: 31.3.2008</td><td>Risiko: hoch</td><td colspan="4">Zeit: 88 h</td></tr>
<tr><td colspan="6">BESCHREIBUNG</td></tr>
<tr><td colspan="6">In SC 10.3.1 wurde ein Konzept zur Personaliserung von Bookmarks erarbeitet. Dieses wird nun anhand der dort getroffenen Designentscheidungen implementiert.</td></tr>
<tr><td colspan="6">Vorbedingung:
• Eine „state-of-the-art“ Analyse wurde durchgeführt, ein Verfahren wurde darauf basierend ausgewählt und in einer Präsentation vorgestellt.
Nachbedingung:
• Die Implementierung der Personalisierung wurde anhand einer Live-Demo und in einer Präsentation vorgestellt.
• Die Live-Demo ist äquivalent zu dem Ablauf des Akzeptanztests der SC 11.3.1.</td></tr>
<tr><td>TASKS</td><td>VERANTW.</td><td>ERL.</td><td>S.G.</td><td colspan="2">S.V.</td></tr>
<tr><td>Implementierung der Personalisierung</td><td>Christian M. & Dennis & Torben & Oliver & Jan</td><td>X</td><td>118</td><td colspan="2">112</td></tr>
<tr><td>Erstellung einer Präsentation zur Personalisierung</td><td>Christian M. & Dennis & Torben</td><td>X</td><td>-</td><td colspan="2">3</td></tr>
</table>

Frontend für kontextuelle Bookmarks

Lfd-Nummer: 19	Priorität: 9	
Frontend für kontextuelle Bookmarks		
Beginn: 12.05.2008	Risiko: niedrig	Zeit: 68 h

Beschreibung

Dem Kunden der Anwendung ist es möglich, seine Plakatdaten in das System einzupflegen. Daher wird eine grafisch ansprechende Anwendung entworfen und implementiert, in der alle Funktionen in drei bis vier Schritten erreicht werden können. Der Kunde kann Bilddaten, dazu passende Services und Metadaten für seine Plakatkampagne über das Frontend einstellen. Diese sind automatisch für jeden Client verfügbar.

Vorbedingung:

- Die endgültige Datenstruktur für kontextuelle Bookmarks muss festgelegt sein.
- Frontend muss Zugriff auf die Datenbank des Olivanda-Projektes besitzen.

Nachbedingung:

- Ein Plakatdatensatz kann in das aktiv laufende System mit Hilfe des Frontends eingebunden werden. Hierzu zählen Bilddaten, kontextuelle Bookmarks sowie genutzte Dienste des Plakaterstellers.
- Die Ergebnisse werden in einer Präsentation und in einer Live-Demo vorgestellt. Die durch das Frontend beispielhaft eingestellte Plakatkampagne wird über den Handy-Client durch Aufnahme des entsprechenden Plakates abgerufen und mit den entsprechenden Bookmarks dargestellt.

Tasks	Verantw.	Erl.	S.G.	S.V.
Recherche nach Realisierungsmöglichkeiten	Jeschua	X	7	4
Technologieauswahl	alle	X	9	4
Konzept einer Datenstruktur	Dennis & Christian M. & Jeschua	X	16	21
Konzept der Benutzeroberfläche	Marcel & Jeschua	X	10	24
Implementierung der Datenstruktur	Jeschua & Dennis & Christian	X	34	29
Implementierung des Frontends	Christian M. & Marcel & Jeschua	X	26	188

Lfd-Nummer: 19	Priorität: 9			
Frontend für kontextuelle Bookmarks (Forts.)				
TASKS	VERANTW.	ERL.	S.G.	S.V.
Datenbankschnittstelle ausfüllen	Christian M. & Jeschua & Marcel	X	4	27
GUI vervollständigen und ausbauen	Christian M. & Jeschua & Marcel	X	30	37

Unbekannte Objekte annotieren

Lfd-Nummer: 20	Priorität: 8	
Unbekannte Objekte annotieren		
Beginn: 12.05.2008	Risiko: mittel	Zeit: 29 h

BESCHREIBUNG
Bisher erhält der Benutzer als Rückmeldung, wenn ein gesendetes Foto zu keinem Plakat zugeordnet werden konnte, lediglich eine diesbezügliche Hinweismeldung. Ziel dieser Storycard ist es, dem Benutzer die Möglichkeit zu geben, Fotos zu annotieren. Hierzu soll er Kampagnen zum Foto auswählen, Dienste zuordnen und das Foto mit Metadaten anreichern können. Diese Daten kann er anschließend persistent in der Datenbank und für andere Nutzer zugänglich speichern.

Vorbedingung:

- Datenstruktur zur Speicherung von Plakaten und Bookmarks muss implementiert worden sein.
- Ein Konzept zur Speicherung und Annotation von zuvor unbekannten Objekten wurde erstellt und in einer Präsentation vorgestellt.

Nachbedingung:

- Das Konzept wurde implementiert und in einer Live-Demo vorgestellt. Hierbei sind zwei verschiedene Fälle zu unterscheiden. Im ersten Fall werden als „unbekannt“ erkannte Plakate zu einer bestehenden oder neu angelegten Kampagne über den Handyclient annotiert. Im zweiten Fall wird zu einem bereits vorhandenen Plakat ein neues Foto hinzugefügt (z.B.: Ein Stuhl von vier verschiedenen Seiten).

TASKS	VERANTW.	ERL.	S.G.	S.V.
Konzepterstellung für die Annotation unbekannter Objekte und Präsentation	Oliver & Jan	X	5	7
Implementierung des Konzeptes zur Annotation unbekannter Objekte	Oliver & Jan	X	44	40

11.3.2. Umsetzung

Erweiterung des Datenbankschemas

Die im Zuge der Entwicklung des Frontends revidierte Datenstruktur ist in Abbildung 11.2 dargestellt.

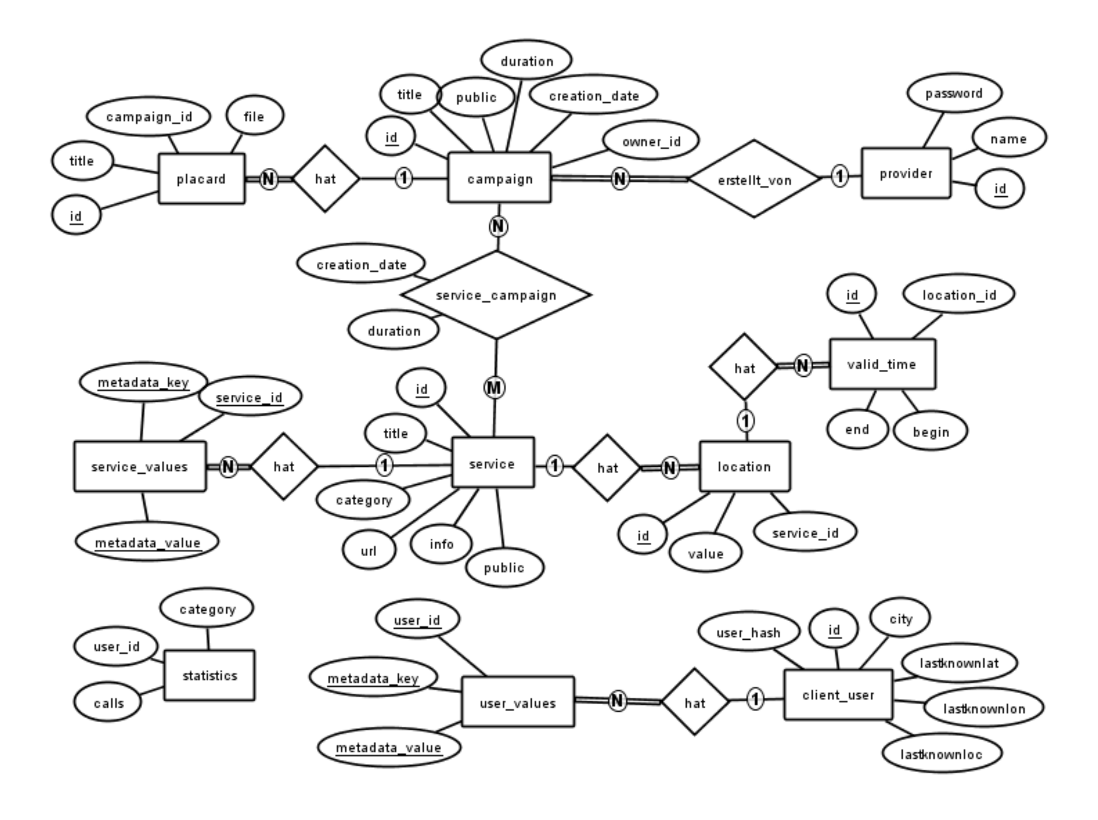


Abbildung 11.2.: ER-Diagramm der Datenbankstruktur nach der Revision

Frontend für kontextuelle Bookmarks

Mit Fortschreiten der Arbeit am Frontend hatte sich herausgestellt, dass der ursprüngliche Ansatz der Seitenführung etwas unglücklich war. Der Benutzer konnte über ein Dropdown-Feld eine Kampagne auswählen und erst danach standen ihm weitere Optionen wie „Dienst“- oder „Plakat zu dieser Kampagne hinzufügen“ zur Verfügung. Dieses Dropdownmenu stand dem Benutzer überall zur Verfügung, damit er bei Bedarf möglichst schnell seine Kampagne wechseln konnte. Das hat aber zu vielen Fehlern im System geführt. So konnte der Nutzer beispielsweise während des Editierens eines Dienstes die Kampagne wechseln, was ihm den Zugriff vom Dienst, den er gerade editiert, entzog.

Hier wurde das Dropdownfeld entfernt und durch eine Liste der eigenen Kampagnen auf der Startseite ersetzt. Der Benutzer kann dort mit einem Klick eine Kampagne auswählen, bearbeiten oder löschen. Wählt er sie aus, werden ihm weitere Details wie die zugehörigen Dienste in einer ähnlichen Listenform, sowie eine Vorschau der zugehörigen Plakate angezeigt. Diese Navigation erschien intuitiver als über ein Dropdownmenu

und weniger fehleranfällig. Der frei gewordene Platz auf der Seite dient dann dazu, dem Benutzer Tipps oder Informationen zu geben. Die aktualisierte Seitenführung ist in Abb. 11.3 zu finden.

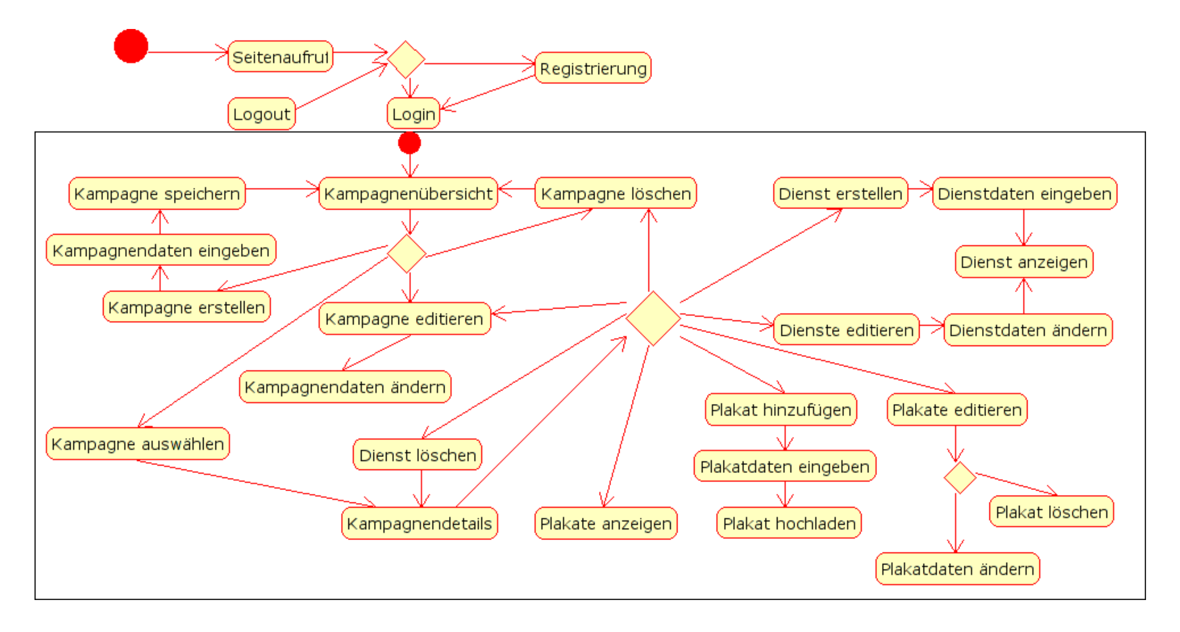


Abbildung 11.3.: Der aktuelle Pageflow des Frontends

Systemarchitektur

Ähnlich der Paketumgestaltung des Servers in der vorletzten Iteration wurden in dieser Iteration die Klassen des Handy-Clients nach Funktionalität in neue Pakete verschoben. Das Paket „connection " enthält alle Klassen zur Client/Server-Kommunikation sowie der Serialisierung bzw. Deserialisierung von XML. „handyui " enthält die Hauptanwendung (Grafische Oberfläche und Logik), „beans " kapselt das Datenmodell des Clients und „musicplayer " die Funktionen zum Abspielen von Musik. Die grundlegende Serverarchitektur wurde hingegen nur um einige Klassen erweitert und der Webzugriff auf den Server in das Paket „WebApp " ausgelagert. Das Klassendiagramm des Clients ist unter 11.5 sowie das Diagramm des Servers unter 11.4 dargestellt.

11.3.3. Ergebnisse

Ergebnisse der Akzeptanztests

Aufgrund von Serverproblemen schlug eine erste Abnahme der Akzeptanztests der Iteration fehl und musste wiederholt werden. Nach Abschluss des zweiten Versuches wurden die Storycards 11.3.1, 11.3.1, 11.3.1 und 11.3.1 als fertiggestellt abgenommen.

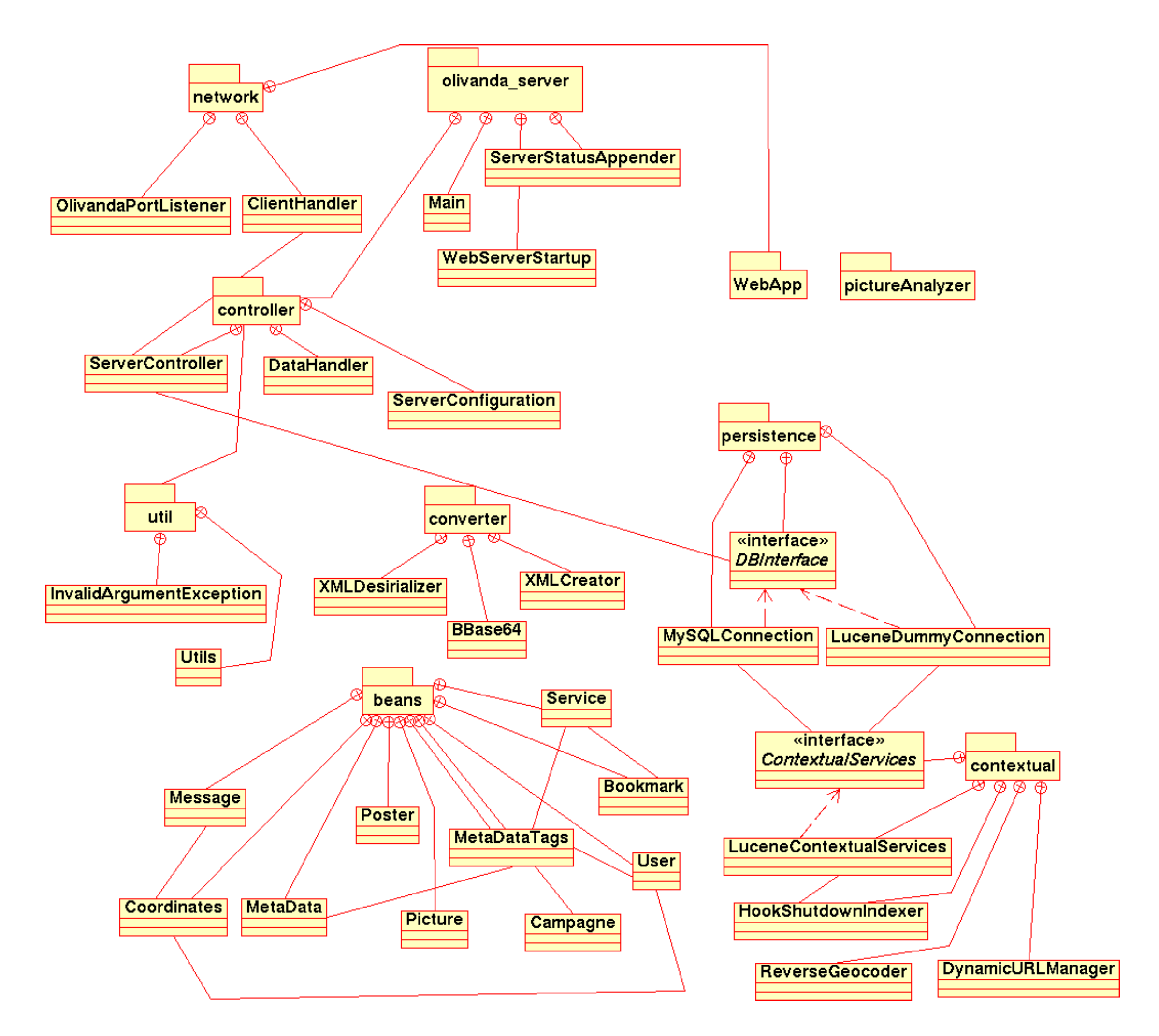


Abbildung 11.4.: Das Aktuelle Klassendiagramm des Servers

Retrospektive

In dieser Iteration überwogen die positiven Eindrücke. Insbesondere führte der beendete Urlaub sehr vieler Projektgruppenmitglieder dazu, dass die Produktivität und damit auch das Gefühl „etwas geschafft zu haben“ im Vergleich zur vorherigen Iteration stark anstieg. Zudem nähern sich viele Aufgaben der entgültigen Fertigstellung.

Die Zusammenarbeit und das Arbeitsklima innerhalb der Gruppe waren sehr gut.

Negativ aufgefallen war in dieser Iteration die enorme Fehleinschätzung des zeitlichen Aufwands der einzelnen Tasks. Insbesondere für die Implementierung des Web-Frontends wurde das Achtfache der geschätzten Zeit benötigt. Durch diese Fehleinschätzung waren die zugeteilten Gruppenmitglieder stärker als vorgesehen ausgelastet und standen für die Bearbeitung anderer Aufgaben nicht zur Verfügung. Dies konnte dadurch etwas

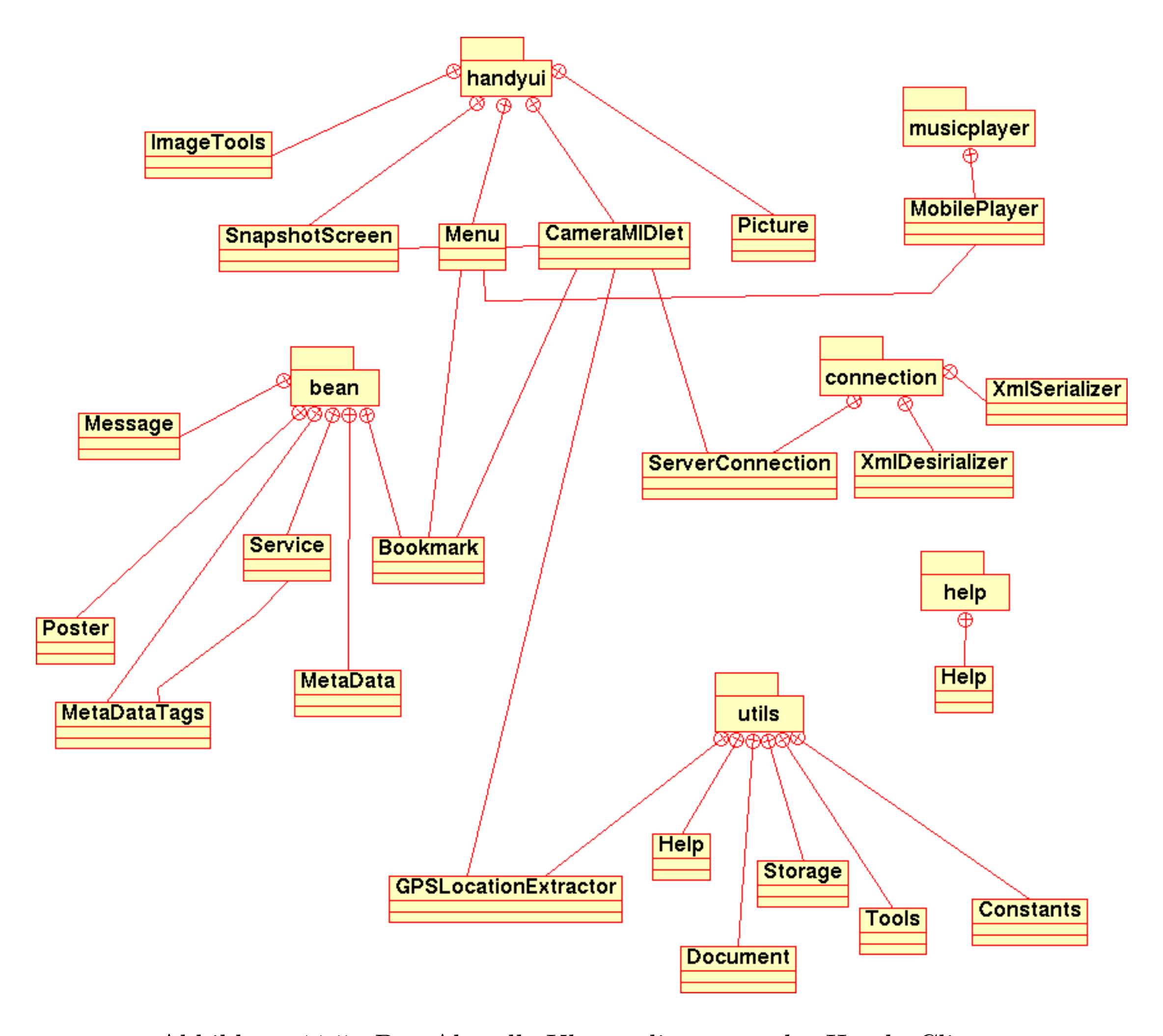


Abbildung 11.5.: Das Aktuelle Klassendiagramm des Handy-Clients

ausgeglichen werden, dass andere Tasks in etwas weniger als der geschätzten Zeit abgeschlossen werden und dadurch die dort eingesetzten Mitglieder die anderen Aufgaben übernehmen konnten.

11.4. Iteration II

Die zweite Iteration des vierten Releases fand in den letzten Wochen der Projektgruppenzeit statt. Hauptaufgaben waren die Fertigstellung des Abschlussberichtes und die Planung der Abschlusspräsentation.

Parallel dazu wurden noch einige gröbere Fehler aus den Programmen behoben, sowie Tests durchgeführt. Neue Features wurden aber nicht mehr eingebaut.

11.4.1. Storycards

Nach Abschluss der meisten Implementierungs-Storycards in der letzten Iteration, wurden nur die zwei Storycards für die Dokumentation aus der letzten Iteration übernommen.

Projekt-Dokumentation

Lfd-Nummer: 1	Priorität: 1	
Projekt-Dokumentation		
Beginn: 01.10.2007	Risiko: niedrig	Zeit: n.a.

BESCHREIBUNG
Die Dokumentation des Projektes wird durch die Verwendung von XP iterativ ergänzt und nach Abschluss des Projektes müssen die verschiedenen Aspekte der Projektarbeit in einer Dokumentation nachlesbar sein. Diese wird online einsehbar sein und die Projekt-Homepage bietet einen aktuellen und optisch ansprechenden Überblick über den Status der Projektarbeit. Der Projekt-Endbericht wird fortlaufend iterativ erweitert.

Vorbedingung:

- n.a.

Nachbedingung:

- Der Abschlussbericht dokumentiert lückenlos alle Aspekte der Projektgruppe über den gesamten Zeitraum der Entwicklung.

TASKS	VERANTW.	ERL.	S.G.	S.V.
Klassendiagramm als rückblickende Übersicht	Jan	X	n.a.	n.a.
Grundgerüst des Endberichts	Christian N.	X	n.a.	n.a.
Umstrukturierung des Endberichts	Marcel S.	X	n.a.	n.a.
Abschlussbericht auf fehlende Textpassagen überprüfen	Jeschua	-	-	3
Fehlende Textpassagen in den Endbericht einfügen	Jeschua	-	-	21
Schlußwort verfassen	-	-	-	-
Client anhand von Screenshots dokumentieren	-	-	-	-
Benutzerhandbuch (Tutorials) erstellen	-	-	-	-
Abschlussbericht auf Schreibfehler überprüfen	Jeschua	-	-	4

Website aktualisieren

Lfd-Nummer: 15	Priorität: 1	
Website aktualisieren		
Beginn: 04.03.2008	Risiko: niedrig	Zeit: n.a.

BESCHREIBUNG
Die Website eines Projektes dient als Aushängeschild und Repräsentation in der Öffentlichkeit und stellt den aktuellen Stand einer Projektarbeit dar. Durch die ständige Weiterentwicklung der Anwendung und damit verbundener Erreichung neuer Releases und Iterationen ist es notwendig, die Daten der Website aktuell zu halten. Die Website bietet den aktuellen Entwicklungsstand der Anwendung sowie eine Darstellung des Projektes (Beschreibung, Zielsetzung, Vorgehensweise und aktuellen Entwicklungsstand) selbst.

Vorbedingung:

- Keine Vorbedingungen.

Nachbedingung:

- Die Webseite zeigt den aktuellen Stand der Entwicklung, eine Beschreibung und die Zielsetzung des Projektes.

TASKS	VERANTW.	ERL.	S.G.	S.V.
Erfassung der Daten, die auf die Website gehören	-	-	-	-
Stand des Projektes auf der Homepage grafisch darstellen	-	-	-	-
Überprüfung der Website auf Aktualität	-	-	-	-

Projektabnahme

Lfd-Nummer: 24	Priorität: 1			
Projektabnahme				
Beginn: 01.09.2008	Risiko: hoch	Zeit: n.a.		
BESCHREIBUNG				
Die Abnahmepräsentation der Projektgruppe Olivanda rückt immer näher (spätestens bis zum 30.09.2008). Im Zuge dessen ist es notwendig, die in den vergangenen Iterationen erstellte Anwendung auf Fehler zu überprüfen, eine Abschlußpräsentation zu erstellen und alle für das Projekt relevanten Daten auf einen Datenträger zu sichern. Hierzu müssen für die Abschlußpräsentation zunächst Entwürfe ausgearbeitet und diese dann weiterentwickelt werden. Die Testfälle für die erstellte Anwendung lassen sich aus dem unter ... definiertem Leitszenario ableiten.				
Vorbedingung: • Keine Vorbedingungen. *Nachbedingung:* • Eine Abschlußpräsentation, die alle relevanten Aspekte der Projektgruppe aufgreift, wurde erstellt und zusammen mit einer Live-Demo der Anwendung vorgestellt. • Die Anwendung wurde auf Fehler anhand des Leitszenarios überprüft und die Ergebnisse in einer Präsentation vorgestellt.				
TASKS	VERANTW.	ERL.	S.G.	S.V.
Entwurf der Anwendungstests	-	-	-	-
Durchführung der Anwendungstests und Präsentation erstellen	-	-	-	-
Ausarbeitung eines Konzeptes zur Abschlußpräsentation	-	-	-	-
Ausarbeitung der Abschlußpräsentation	-	-	-	-
Sicherung aller projektrelevanten Daten	-	-	-	-

Abschluß

In dieser Iteration wurden lediglich „Restarbeiten“ an der Anwendungsimplementierung vorgenommen und der Fokus auf die Abschlußpräsentation der Projektgruppe sowie die Ausarbeitung des Endberichtes gesetzt. Somit wurden das Kapitel „Umsetzung“ sowie die zur Iteration gehörende Retrospektive ausgelassen.

12. Anhang

12.1. Installation

Im Nachfolgenden wird detailliert auf die Installation des gesamten Olivanda-Projektes Schritt für Schritt eingegangen. Als Referenzsystem für den Server bzw. den Handyemulator dient eine Microsoft Windows XP Installation sowie ein Nokia N95 Handy nach dem Series 60r3 Standard. Der Server lässt sich plattformunabhängig ebenfalls unter Linux, FreeBSD sowie Mac OSX betreiben.

Die Handysoftware lässt sich momentan aufgrund des Einsatzes von J2MEPolish und Sun Java lediglich erfolgreich unter Linux und Microsoft Windows kompilieren.

12.1.1. Benötigte Technologien

Alle benötigten Technologhien finden sich für Windows auf der Installations DVD.

- MySQL ab Version 5.0 für die Datenbank (z.B. in XAMPP enthalten)
- Webserver mit PHP 5 für das Webfrontend (z.B. in XAMPP enthalten)
- Java JDK ab der Version 1.6
- J2MEPolish in der Version 2.0.4
- Netbeans IDE ab Version 6.1 (das Paket mit den Plugins für die Mobilentwicklung)
- Sun WTK 2.5.X als Handyemulator
- Optional: XAMPP zur vereinfachten Datenbank- und Webadministration

Java JDK 1.6 installieren

Java homepage: java.sun.com

- Laden Sie sich das Java JDK 1.6 von der Sun Homepage runter.
- Starten Sie die Jar datei mit dem Komando: „java -jar dateiname.jar“ und folgen Sie den Installationsanweisungen.

Xampp installieren

Xampp homepage: http://www.apachefriends.org/de/xampp.html

- Laden Sie sich eine Xampp version runter und installieren Sie diese.
- Starten Sie die Xampp-Control.exe und starten Sie den Apache und Mysql.
- Klicken Sie auf Exit.

12.1.2. Anwender

Server mit Setup installieren

Das Setup befindet sich in der setup.zip-Datei auf der DVD im Setupordner. Diese Datei müssen Sie zunächst entpacken. Danach führen Sie die enthaltende Datei Olivanda_Server.jar per Doppelklick aus. Alternativ können Sie per Kommandozeile in das entsprechende Verzeichnis wechseln und dort mit dem Befehl java -jar Olivanda_Server.jar starten. Es ist von immanenter Bedeutung, dass das aktuelle Arbeitsverzeichnis der VM tatsächlich das Setupverzeichnis ist, da ansonsten die Setup-Daten nicht gefunden werden können und die Installation somit nicht durchgeführt werden kann.

Nachdem dies geschehen ist, erfolgt die Installation und das Setup über eine geführte Weboberfläche. Dies erlaubt ebenfalls das komfortable Installieren auf einem nicht GUI-fähigen Remoteserver, Sie müssen dazu nur einen Port in dem Bereich von 8050 bis 9000 in Ihrer Firewall freigeben. Beachten Sie, dass das Setup einen Port aus genau diesem Bereich benötigt und eine Installation ansonsten nicht erfolgen kann. Wenn Ihr System eine GUI-Oberfläche inklusive Systemwebbrowser besitzt, öffnet sich automatisch Ihr Standard Webbrowser mit der korrekten Seite. Wenn dies nicht geschieht kontrollieren Sie bitte die Konsolenausgabe und geben den vom Setup automatisch ermittelten nächsten freien Port in dem Webbrowser Ihrer Wahl an.

Wenn Sie dem Setup-Assistenten folgen, müssen Sie als erstes den Pfad für das PHP-Frontend angeben, dass sich konsequenterweise im Root-Verzeichnis eines funktionstüchtigen Webservers befinden sollte. Wählen Sie immer das Root-Verzeichnis, da dort immer automatisch die Subverzeichnisse olivanda/web erzeugt werden, die für den Einsatz mit dem Handy auch zwingend erforderlich sind. Das Serververzeichnis ist hingegen beliebig und kann nach Gutdünken installiert werden. Beachten Sie bitte, dass der Server allerdings Schreibzugriff in seinem Arbeitsverzeichnis benötigt, da dort die Log-Dateien hinterlegt werden.

Im zweiten Schritt sind die wesentlichsten unbedingt nötigen Konfigurationsdaten zu hinterlegen. Wenn Sie unsicher bezüglich bestimmter Daten wie Server-Port oder Datenbankinformationen sind, kontaktieren Sie bitte Ihren Administrator. Der Serverport spezifiziert den Port der Administrationsoberfläche für den Java-Server, der ebenfalls vom PHP-Frontend erreichbar sein muss, da diese für die Steuerung notwendig sind.

Ebenfalls frei verfügbar muss der Port 27211 sein, auf dem sich das Handy später verbinden wird. Für nähere Beschreibungen besitzt jedes Feld einen Tooltip, der diesen weiter beschreibt.

Wichtig ist noch anzumerken, dass die Verzeichnisse des Servers und des PHP-Frontends vorher gelöscht werden um eine völlig saubere Umgebung herzustellen. Sichern Sie bitte somit ältere Installationen sofern die Verzeichnisse weitere Daten enthalten.

Wenn alles erfolgreich durchgeführt wurde, können Sie automatisch den Java-Server starten und er beginnt automatisch mit der Indexierung der vorhandenen Default-Daten der Datenbank und der Referenzbilder.

Viel Spaß

Client auf Nokia N95 installieren

- Öffnen Sie den Order /setup/N95 auf der DVD. Dort liegen die beiden Dateien Nokia-N95-de_DE-olivanda.jad und Nokia-N95-de_DE-olivanda.jar die auf das Nokia N95 Handy kopiert werden müssen.
- Schließen Sie nun das N95 mit dem mitgelieferten USB-Kabel an Ihren Rechner an. Daraufhin erscheint auf dem N95 eine Nachfrage wie es verbunden werden soll. Wählen Sie hier Datentransfer aus. Danach wird Ihr Betriebssystem einen weiteren Wechseldatenträger erkennen.
- Wechseln Sie in das Wurzelverzeichnis dieses Datenträgers und kopieren Sie die beiden genannten Dateien hinein.
- Trennen Sie nun das N95 von Ihrem Rechner. Benutzen Sie wenn möglich die Datenträger aushängen Funktion Ihres Systems um potentiellen Datenverlust zu vermeiden.
- Nehmen Sie nun Ihr N95 und wechseln Sie in das Hauptmenü Ihres Handys. Benutzen Sie dazu die namenlose Taste mit dem blau-gefärbten Orbitalmodell ähnlichen Symbol, verwandt mit dem Recycling-Symbol der dualen Abfallwirtschaft.
- Dort in den Punkt System wechseln
- Dann den Dateimanager auswählen
- Hier drücken Sie dann anschließend auf die Navigationstaste rechts um auf die Speicherkarte des Handys zuzugreifen.
- Wählen Sie nun das jar oder jad-File um die Installation zu starten
- Wählen Sie als Installationsort den Telefonspeicher
- Das nun installierte Olivanda Programm finden Sie im Hauptmenü unter Programme wieder
- Nachdem Sie die Anwendung gestartet haben, vergessen Sie bitte nicht den korrekten Server auszuwählen

Frontend Anwendung

- Installatieren und starten Sie den Olivanda Server.
- Starten Sie einen Webbrowser und geben Sie die URL http://[serverhost/][evtl. unterverzeichnis/]olivanda/web ein.
- Es erscheint die Loginseite auf der Sie sich einloggen oder registrieren können.
- Nach erfolgreichem Login haben Sie die Möglichkeit Kampagnen, Plakate und Dieste zu verwalten.

Server Frontend Anwendung

- Server Funktionen:
  - Serverumgebung zeigt den Systemstatus an. Darunter zu finden sind Speicherverbrauch, Arbeitsverzeichnis und die Anzahl der indexierten Bilder.
  - Unter Serverkonfiguration stehen Ihnen sämtliche verfügbaren Optionen des Servers zur Verfügung. Für nähere Erläuterungen beachten Sie bitte die entsprechenden Tooltipps. Beachten Sie ferner, dass Sie das PHP-Frontend niemals direkt konfigurieren sollten, da das Serverfrontend eine automatische Synchronisierung vornimmt.
- Server-Status:
  - Zeigt alle aktuellen Logausgaben des Servers, der aktuell laufenden Instanz an
  - Wenn Sie die Log-Dateien älterer Instanzen betrachten möchten, finden Sie diese im Arbeitsverzeichnis der Anwendung
- Downloads:
  - Laden Sie hier die vorkompilierten Binaries für den Emulator oder das Nokia N95 Mobiltelefon herunter.

Client Annotation

- Installieren Sie den Olivanda Client auf dem Nokia N95 und starten Sie die Anwendung.
- Wählen Sie den Punkt Konfiguration und stellen Sie die IP Adresse des Olivandaservers ein.
- Wählen Sie „Foto machen“ und fotografieren Sie ein Bild das nicht erkannt werden soll

- Beim Versenden des Bildes müssen Sie ggf. noch Einstellungen bezüglich Ihrer Internetverbindung auswählen.
- Nachdem das Bild nicht erkannt wurde, können Sie den Punkt „Foto eintragen“ auswählen um das Foto zu annotieren.
- Es öffnet sich der Handybrowser und Sie können entweder eine Kampagne suchen der Sie das Bild annotieren wollen oder Sie erstellen eine neu Kampagne mit der das Bild annotiert wird.
- Anschließend klicken Sie auf fertigstellen. Das Bild wird nun annotiert.
- Sie bekommen eine Anzeige ob das Bild analysiert wurde und können auf Beenden klicken.
- Wählen Sie nun den Zurück-button um wieder ins Hauptmenü zu gelangen.

Client Anwendung

- Installieren Sie den Olivanda Client auf dem Nokia N95 und starten Sie die Anwendung.
- Wählen Sie den Punkt Konfiguration um stellen Sie die IP Adresse des Olivandaservers ein.
- Wählen Sie „Foto machen“ und fotografieren Sie ein Bild das erkannt werden soll
- Wird das Bild erkannt, bekommen Sie einen Bookmark mit Diensten zurück.
- Wird das Bild nicht erkannt haben Sie die Möglichkeit das Bild zu annotieren oder ein neues Foto zu machen.
- Unter dem Punkt „Meine Bookmarks“ finden Sie alle Bookmarks die Sie bisher bekommen haben.

12.1.3. Entwickler

Netbeans 6.1 Installation

Laden Sie sich Netbeans6.1 von http://www.netbeans.org herunter und installieren Sie dieses. Um sicherzugehen, dass wirklich alle benötigten Entwicklermodule enthalten sind, ist die vollständige Netbeansversion (ca. 200Mb Download) zu empfehlen.

J2ME-Polish installieren

- Laden Sie sich J2ME Polish 2.0.4 von der Homepage http://www.j2mepolish.org/ runter.

- Starten Sie die Installationsdatei und klicken Sie auf next.
- Akzeptieren Sie die Lizenzbedingungen und klicken Sie auf next.
- Wählen Sie ein Installationsverzeichnis aus und klicken auf next.
- Wählen Sie als Verzeichnis des WTK im Netbeanshomeverzeichnis /mobility8/WTK2.5.2 aus.
- Wählen Sie ganz unten das netbeans home aus (z.B. c:/Programme/netbeans6.1)
- Klicken Sie auf next.
- Wählen Sie die Checkbox bei Netbeans Integration aus und klicken Sie auf next.
- Klicken Sie auf Quit.

Serverinstallation unter Netbeans

- Starten Sie Netbeans und wählen Sie „new Projekt“
- Wählen Sie als Kategorie Java und unter Projekts „Java Projekt with Existing Sources“ und klicken Sie auf next
- Wählen Sie einen Projektnamen und klicken Sie auf next.
- Klicken Sie unter Source Package Folders auf Add Folder und wählen Sie InstallationsDVD/source/core/Olivanda_Server/src aus.
- Klicken Sie unter Test Package Folders auf Add Folder und wählen Sie InstallationsDVD/source/core/Olivanda_Server/test aus.
- Klicken Sie auf next
- Klicken Sie auf Finish
- Klicken Sie mit der rechten Maustaste auf Ihr Projekt und wählen Sie den Punkt Properties aus.
- Wählen Sie den Punkt Libraries aus und klicken Sie auf Add JAR/Folder und wählen Sie InstallationsDVD/source/core/Olivanda_Server/lib aus und fügen Sie alle Pakete aus.
- Klicken Sie auf OK.
- Nun können Sie das Projekt kompilieren und starten

Client in Netbeans laden

J2ME Polish homepage: http://j2mepolish.org

- legen Sie ein neues Projekt an

- wählen Sie das Mobility und J2ME Polish Project aus
- wählen Sie „Blank“ aus
- öffnen Sie die Properties des Projektes und stellen Sie sicher das folgende Pakete eingebunden werden:
  - File Connection and PIM Option Packages 1.0
  - Location Based API 1.0.1
  - Mobile Media API 1.1
- unter Properties - Manage Emulators - Tool&Extentions - open Preferences - Securety folgende Einstellungen vornehmen:
  - Securety Policy = JTWI
  - Securety domain = trusted
- Resources und Sources von der Olivanda DVD in den Projektordner kopieren.
- In der „build.xml“ kann nun die Startdatei für das Projekt festgelegt werden. Dazu muss die Zeile

  ```
  <midlet class="de.enough.polish.sample.accessible.MenuMidlet"
          name="Accessible" />
  ```

  durch

  ```
  <midlet class="handyui.CameraMIDlet" name="Olivanda" />
  ```

  ersetzt werden. Der unter dem Attribut „name“ angegebene Bezeichner gibt dabei an, wie die Anwendung später auf dem Handy benannt sein soll. Soll zum Testen der Standard-WTK-Emulator verwendet werden, ist nun noch ein Eintrag im „build“-Tag nötig. Dazu muss ein zusätzliches Attribut „binarylibraries“ mit dem Parameter „location.jar“ hinzugefügt werden, so dass das öffnende Tag folgenden Aufbau hat:

  ```
  <build binarylibraries="location.jar"
          symbols="ExampleSymbol, AnotherExample"
          fullscreen="menu" usePolishGui="true"
          workDir="${dir.work}"
          compilerMode="${enableCompilerMode}"
          compilerDestDir="build/classes"
          compilerModePreverify="true">
  ```

  Wird dieser Eintrag nicht vorgenommen, lässt sich die Anwendung auf dem Emulator nicht starten. Sehr wichtig ist, dass das „binarylibraries“-Attribut wieder entfernt werden muss, wenn das Projekt für das N95 kompiliert werden und die Anwendung dort GPS-Daten auslesen soll. Bleibt der Eintrag bestehen, ist das Handy nicht in der Lage, seine vorinstallierte Location-API zu nutzen.

Literaturverzeichnis

[1] Auclair, A. u. a.: How to Use SIFT, Vectors to Analyze an Imange with Database Templates. In: *Conference Adaptive Multimedia Retrieval, Paris* (2007)

[2] Bay, H. u. a.: SURF: Speeded Up Robust Features. In: *European Conference on Computer Vision* (2006)

[3] Bentley, J. L.: K-D Trees for Semidynamic Point Sets. In: *Proceedings of the sixth annual symposium on Computational geometry* (1990)

[4] Bicego, M. u. a.: On the Use of SIFT, Features for Face Authentication. In: *Conference on Computer Vision and Pattern Recognition Workshop* (2006)

[5] Fischler, M. A. ; Bolles, R. C.: Random Sample Consensus: A Paradigm for Model Fitting with Appilcations to Image Analysis and Automated Cartography. In: *Commun. ACM* (1981)

[6] Gionis, A. u. a.: Similarity Search in High Dimensions via Hashing. In: *Proceedings of the 25th VLDB Conference, Edinburgh, Scotland* (1999)

[7] Gordon, I. ; Lowe, D. G.: Scene Modelling, Recognition and Tracking with Invariant Image Features. In: *Third IEEE and ACM International Symposium on Mixed and Augmented Reality* (2004)

[8] Harris, C. ; Stephens, M. : A combined corner and edge detector. In: *Proceedings of the 4th Alvey Vision Conference* (1988)

[9] Henze, N. u. a.: Contextual Bookmarks. In: *Mobile Interaction with the Real World Workshop @ MobileHCI 07* (2007)

[10] http://www.tbray.org/ongoing/When/200x/2006/12/21/JSON: *JSON and XML*. http://www.tbray.org/ongoing/When/200x/2006/12/21/JSON. Version: Dezember 2006

[11] Ke, Y. u. a.: Efficient Near-duplicate Detection and Sub-image Retrieval. In: *Proceedings of ACM International Conference on Multimedia (MM)* (2004)

[12] Ke, Y. ; Sukthankar, R. : PCA-SIFT,: A More Distinctive Representation for Local Image Descriptors. In: *IEEE Conf. on Computer Vision and Pattern Recognition* (2004)

[13] Khassanov, A. u. a.: Automatic Panorama Recognition. In: *School of Computer Science McGill University, Montreal, QC, Canada* (2004)

[14] Lehmann, T. u. a.: *Bildverarbeitung für die Medizin.* Berlin : Springer, 1997

[15] Lowe, D. ; Little, J. : Vision-based mobile robot localization and mapping using scale-invariant features. In: *IEEE International Conference on Robotics and Automation* (2001)

[16] Lowe, D. G.: Object Recognition from Local Scale-Invariant Features. In: *ICCV* (1999)

[17] Lowe, D. G.: Local Feature View Clustering for 3D Object Recognition. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2001)

[18] Lowe, D. G.: Distinctive Image Features from Scale-Invariant Keypoints. In: *International Journal of Computer Vision* (2004)

[19] Mikolajczyk, K. ; Schmid, C. : A Performance Evaluation of Local Descriptors. In: *IEEE Trans. Pattern Anal. Mach. Intell.* (2005)

[20] Smeulders, A. W. u. a.: Content-Based Image Retrieval at the End of the Early Years. In: *IEEE Transactions On Pattern Analysis And Maschine Intelligence* (2000)

[21] Tanaka, K. : Mechanisms of visual object recognition: monkey and human studies. In: *Current Opinion in Neurobiology* (1997)

[22] Wikipedia: *Recall und Precision.* `http://de.wikipedia.org/wiki/Recall_und_Precision`. Version: August 2008

[23] Williams, L. A.: THE COLLABORATIVE SOFTWARE PROCESS. In: *Comput. Vision Graph. Image Process.* (2000)